1

20001013182

AUTOMATING
THE SATELLITE RANGE SCHEDULING PROCESS

THESIS

Timothy D. Gooley
Captain, USAF

AFIT/GOR/ENS/93M-06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/93M-06

AUTOMATING
THE SATELLITE RANGE SCHEDULING PROCESS

THESIS

Timothy D. Gooley
Captain, USAF

AFIT/GOR/ENS/93M-06

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE March 1993	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE AUTOMATING THE SATELLITE RANGE SCHEDULING PROCESS			5. FUNDING NUMBERS
6. AUTHOR(S) Timothy D. Gooley			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GOR/ENS/93M-06
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Phillips Laboratory, Space Software Technology Division, PL-VTES, KIRTLAND AFB NM 87117-5776			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Distribution Unlimited			

93-06908

93 4 02 067

13. ABSTRACT (*Maximum 200 words*)

Satellite range scheduling is a complex problem that involves scheduling satellite supports in which a satellite and a specific remote tracking station are assigned a time window during which they communicate with each other. As the number and complexity of satellite supports continue to increase, more pressure is placed on the current manual system to efficiently generate a schedule. The objective of this research was to develop a methodology that will automate the generation of the initial 24 hour schedule. The goal of the algorithm developed was to schedule as many conflict free supports as possible. A two phased approach was developed to schedule the supports. The first phase scheduled as many low altitude satellite supports as possible, while the second phase scheduled as many additional high altitude satellite supports as possible. For both phases, schedule generation and schedule improvement algorithms were developed. The schedule generation algorithms are a mixed integer program linking procedure and an insertion procedure. The schedule improvement algorithms are a two satellite interchange procedure and a three satellite interchange procedure. A schedule was generated for six representative data sets with encouraging results. At least 91% of all satellite support requests were scheduled for each day. These results were comparable to results of the current range schedulers and a previous automation study. Based on the results reported, the methodology presented in this research effort seems to be a valid approach for automating the initial 24 hour schedule.

14. SUBJECT TERMS			15. NUMBER OF PAGES
Scheduling, Mixed Integer Programming, Insertion Procedure, Interchange Procedure			240
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THESIS APPROVAL

STUDENT: Capt Timothy D. Gooley CLASS: GOR-93M

THESIS TITLE: Automating the Satellite Range Scheduling Process

DEFENSE DATE: March 8, 1993

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Co-Advisor	Maj John J. Borsi/ENS	
Co-Advisor	Lt Col James T. Moore/ENS	James T. Moore

AFIT/GOR/ENS/93M-06

AUTOMATING THE SATELLITE RANGE SCHEDULING PROCESS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Timothy D. Gooley, B.S. Operations Research, MBA
Captain, USAF

March, 1993

Approved for public release; distribution unlimited

Preface

In developing the methodology for this research effort, as well as writing the thesis document, I had a great deal of assistance from others. I would first like to express my gratitude and appreciation to my advisors, Maj John Borsi and Lt Col James Moore , for their insight and guidance. I would also like to thank my sponsor, Phillips Laboratory, for their interest and the funding provided for this research effort. Additionally, I would like to thank Ken Chambers and Francis Wong of the 21 SOPS, 1Lt Greg Schultz, satellite range scheduling project officer, and John List, Paramax Systems Corporation, for their technical assistance.

Most importantly, I would like to thank my wife, Karen, and two sons, Jonathan and Joshua, for their love and support during my AFIT experience. Finally, I would like to thank the Lord for his love and answers to my numerous prayers during this research effort.

Timothy D. Gooley

Table of Contents

List of Figures

List of Tables

AFIT/GOR/ENS/93M-06

Abstract

Satellite range scheduling is a complex problem that involves scheduling satellite supports in which a satellite and a specific remote tracking station are assigned a time window during which they communicate with each other. As the number and complexity of satellite supports continue to increase, more pressure is placed on the current manual system to efficiently generate a schedule. The objective of this research was to develop a methodology that will automate the generation of the initial 24 hour schedule. The goal of the algorithm developed was to schedule as many conflict free supports as possible. A two phased approach was developed to schedule the supports. The first phase scheduled as many low altitude satellite supports as possible, while the second phase scheduled as many additional high altitude satellite supports as possible. For both phases, schedule generation and schedule improvement algorithms were developed. For low altitude satellites, the schedule generation algorithm applied a mixed integer program with a linking procedure, and the schedule improvement algorithm was a two satellite interchange procedure. For medium/high altitude satellites, the schedule generation algorithm was an insertion procedure and the schedule improvement algorithm was a three satellite interchange procedure.

A schedule was generated for six representative data sets with encouraging results. At least 91% of all satellite support requests were scheduled for each day. These results were comparable to results of the current range schedulers and a previous automation study. Based on the results reported, the methodology presented in this research effort seems to be a valid approach for automating the initial 24 hour schedule.

AUTOMATING THE SATELLITE RANGE SCHEDULING PROCESS

I. Introduction

Overview

Satellites and space technology have become increasingly important in the last decade. Annual sales of space products in the United States have increased from approximately eight billion dollars to thirty-two billion dollars over the last ten years (Asker:1991,51). The U.S. government has been a leader in space expansion as its purchases account for 90% of all U.S. space product sales over this time period (Asker:1991,51). Accordingly, space systems are playing a more vital role in our national defense as the surveillance, warning, and communications capabilities offered by these systems have become more important in the mixture of U.S. defense systems. Satellites placed in orbit to provide these capabilities generally require frequent contact with remote tracking stations for the transmission and receipt of information required for mission accomplishment and continued satellite operations. This thesis focuses on these satellite/remote tracking station contacts.

The Air Force Satellite Control Network (AFSCN) manages communications between satellites and remote tracking stations. The AFSCN is comprised of four major elements : 1) satellites, 2) Mission Control Complexes (MCCs), 3) remote tracking stations (RTSs) and 4) Resource Control Complexes (RCCs). For the purposes of this thesis, **satellite** refers to a man made earth orbiting vehicle. A MCC contains the personnel that maintain operational control over the satellite and determine what commands should be transmitted to the satellite. Operational control entails planning for future satellite communications, ensuring current communication requirements are met and analyzing the data sent from the satellite. The RTSs are located throughout the world and contain the personnel that control the equipment which communicates with the satellite. The RTSs provide a necessary link between the MCC and the satellite, enabling MCC personnel to control the satellite and receive information from it. The RCCs oversee the entire AFSCN and provide a centralized location where coordination between MCC and RTS personnel can take place. Each of the elements of the AFSCN will be discussed later in this chapter.

Satellites must transmit or receive information in a timely manner to perform efficiently. Therefore, a scheduling process is required to determine **when** (the time) and **where** (remote tracking station location) essential information will be transmitted between a satellite and a remote tracking station. A **satellite support** is any transmission of data/information between a satellite and a remote tracking station. In general, satellite range scheduling (SRS) involves scheduling satellite supports in which a satellite and a specific remote tracking station are assigned a time window during which they communicate with each other. Generally, there are two requirements for a support to occur: 1) the satellite must be within the line of sight of the remote tracking station and 2) the antenna at the remote tracking station must be pointed in the direction of the satellite.

There are several different types of supports. Some of the major support types are: 1) Commanding - which tells the satellite to perform a specific function, 2) Health and Status - which insures the satellite is operating correctly, and 3) Payload - the satellite transmits payload data to the remote tracking station. The amount of time required for the satellite support varies depending on the type of support required and the amount of information transferred between the satellite and the RTS. An understanding of each of the components of the AFSCN will provide further insight into the role each component plays in the scheduling process.

Background

Satellites in the AFSCN. The following discussion will briefly describe the types of satellites that are part of the AFSCN. A majority of the satellites that are part of the military space program as well as some government sponsored satellites are controlled through the Air Force Satellite Control Network (AFSCN). Currently, the AFSCN controls up to 90 different satellites (Chambers, 17 June 1992). In general, AFSCN supported satellites may be categorized by the altitude (low,medium, or high) at which they orbit. In addition, the AFSCN provides launch support for ballistic missiles and assent only support for launches of satellites not normally under the control of the AFSCN.

Low altitude satellites are characterized by near polar orbits and have orbital altitudes of approximately 100 to 200 nautical miles. These satellites have the shortest pass durations (defined as the amount of time a satellite is visible from a single point on the earth), between 2.5 and 10 minutes, and require contact with an average of 1.5 remote tracking stations per revolution (AFSCF Training Manual Course No. 203,1983:1-7) . **Medium altitude satellites** have an orbital inclination near 90 degrees and have orbital altitudes between 200 and 1000 nautical miles. Medium altitude satellites have expected pass durations of 10 to 20 minutes and normally require one remote tracking station contact every two orbits (AFSCF Training Manual Course No. 203,1983:1-7). **High altitude satellites** are characterized by orbital altitudes in excess of 10,000 nautical miles. Orbital altitudes between 1000 and 10,000 nautical miles contain mostly space debris and therefore no satellites in the AFSCN orbit in this altitude range. The visibility window for high altitude satellites can range between one hour and continuous visibility. A continuously visible satellite is always visible to at least one RTS but is not always visible to a single RTS. The satellite support length for high altitude sateliites can vary from five minutes to several days. (AFSCF Training Manual Course No. 203,1983:1-7). In addition to supporting satellite on-orbit operations, AFSCN supports **ballistic**

missile launches. The network provides tracking and telemetry support for these launches with their support times ranging between ten and thirty minutes (AFSCF Training Manual Course No. 203,1983:1-8). The last class of satellite support is **assent only support**, which entails supporting launches and assents for satellites that are not otherwise under the control of the AFSCN (AFSCF Training Manual Course No. 203,1983:1-8).

Mission Control Complex (MCC). As stated earlier, a MCC contains the personnel and equipment that maintain operational control of one or more satellite programs and is located at a resource control complex. A satellite program contains satellites that perform similar missions. The MCCs are responsible for providing input to the satellite range schedulers on the support requirements for their satellites as well as listing the remote tracking stations capable of providing the requested support.

Remote Tracking Station (RTS). The remote tracking stations are located throughout the world and contain the personnel and equipment that communicate with the satellites. There are nine RTSs located throughout the world. Five of the RTSs can support two satellites simultaneously and one can support three satellites simultaneously; thus, up to sixteen satellites can be supported at any one time. Because the communication path from a RTS to a satellite is essentially line of sight, the amount of time a satellite is visible depends on several factors, with the altitude, speed and position of the satellite, and the rotation of the earth being the key factors. RTSs have several pieces of equipment necessary to communicate with satellites. Antenna systems are needed for tracking the position of the satellite, sending information to or receiving information from the satellite, and communicating between remote tracking stations. In fact, no communication can occur between the satellite and the RTS until the antenna is pointed in the direction of the satellite. Information received or sent by the antenna systems must be stored and processed by various computer communication systems. The equipment requirements may vary depending on the type of support (commanding or health and status) required.

Resource Control Complexes (RCCs). The RCCs contain both the schedulers who generate the satellite support schedule and the MCC personnel who establish satellite support requirements. The AFSCN has two Resource Control Complexes: the Consolidated Space Test Center (CSTC) located at Onizuka AFB in California and the Consolidated Space Operations Center (CSOC) located at Falcon AFB in Colorado. The satellite support schedule for the AFSCN is developed jointly by schedulers at both locations.

Satellite Range Scheduling Process

Overview. Developing a schedule for a single day is a two week process that culminates in a conflict free schedule for each day. A conflict free schedule exists when 1) no RTS is scheduled to communicate with more satellites than it can support at any one time, and 2) each RTS has sufficient setup time between supports. Obviously, schedules for several days are being developed at the same time. Inputs to the schedule come primarily from the MCCs and RTSs. The MCCs provide all the required information on satellite support requirements via a weekly program action plan (PAP). RTS personnel provide downtime information on routine or major maintenance of the RTS. In general, the schedulers schedule from the most restrictive supports to the least restrictive supports. For example, a very restrictive support would be for a low altitude satellite that is visible to only one RTS for a short time period. A very flexible support would be for a high altitude satellite visible to several RTSs for a very long time period. In general, restrictive supports are scheduled once their time window is known and the flexible supports are shuffled to make room for the restrictive supports.

Previously, the schedulers at the CSTC generated the schedule manually and displayed it on an 84 foot paper acquisition chart. This chart contained the entire schedule for a one week period. The acquisition chart showed the time period and remote tracking station for every scheduled satellite support as well as any additional information the scheduler required. The vertical axis of the chart represented all the different RTSs and was divided into horizontal sections, with each section representing a RTS. Time lines were shown on the horizontal axis. Pieces of colored tape were utilized to designate a satellite support and assigned a location on the chart at a specific RTS and time period. The chart contained additional hand written information such as type of support requested and any special information the RTS or scheduler needed. Each satellite program was assigned a unique pattern of tape with each satellite in that program given a different color. Non-flight supports, such as RTS maintenance, were assigned pieces of yellow tape and were also indicated on the chart. Recently, an electronic acquisition chart was developed as part of the Automated Scheduling Tools for Range Operations (ASTRO) computer system. The electronic chart displays the acquisition chart on a large screen computer monitor. The layout and symbols used on the paper acquisition chart are replicated on the electronic acquisition chart. The ASTRO system was specifically developed to automate a section of the SRS problem and includes, in addition to the electronic acquisition chart, a database, the capability to use voice activated commands to adjust the schedule, and comprehensive report generation capability.

The scheduling process can be divided into four primary phases: 1) construction of the "seven day" schedule, 2) development of the initial 24 hour schedule, 3) conflict resolution, and 4) real time scheduling. Each phase of the scheduling process is explained below.

Seven Day Schedule. The seven day schedule construction consists of 1) collecting all possible inputs, such as known support requirements, launch requests or non-flight requirements (like RTS maintenance), 2) building an initial schedule and

3) publishing this schedule. The seven day schedule is a schedule for one day that is generated one week in advance and includes only satellites whose visibility and support requirements are known a week in advance. Therefore, usually only high altitude satellites are included in this schedule. The satellites are then randomly assigned support times and RTS with little time spent on minimizing conflicts. A support message is then printed to each MCC and RTS indicating the support times and requirements.

Initial 24 Hour Schedule. After the seven day schedule is completed, a schedule which includes all possible satellite supports is constructed. The initial 24 hour schedule is developed to the point where coordination with MCC/RTS personnel is required to resolve any remaining conflicts. The construction of this schedule is currently an iterative process. The reason for this is that the support and visibility times for low altitude and other time critical satellites are not known until a few days before the day of the support. Therefore, the schedulers start generating a schedule with known satellite requests. Low altitude satellites are scheduled when their support requirement and visibility times become known. This usually means adjusting the schedule to make room for the new support. The criteria for moving satellite supports is based on the flexibility of the satellite support requirement. The high altitude satellite supports are usually more flexible and are commonly moved to make room for the low altitude satellite supports. Therefore, the low altitude satellite supports generally have scheduling priority. However, a medium or high altitude satellite will have scheduling priority when the satellite support is mission critical.

Additionally, any known downtimes for the RTS must be incorporated into the initial 24 hour schedule. RTS downtime falls into one of two general categories: routine maintenance or major maintenance/modification. Major RTS maintenance or modification, possibly due to equipment malfunction, renders an RTS incapable of supporting satellite contacts. Therefore, satellite contacts cannot be scheduled during these downtimes. On the other hand, in some cases, the time for routine maintenance can be rescheduled to allow satellite contacts.

Conflict Resolution Phase. The conflict resolution phase begins after all possible supports have been scheduled. There are essentially two types of conflicts: a visibility conflict and a turnaround conflict. A visibility conflict exists when a satellite can only be supported at times in which it's visible RTS is scheduled to communicate with another satellite. A turnaround conflict exists when a satellite support cannot be scheduled due to insufficient time for a RTS to reconfigure between supports. Reconfiguring includes pointing the antenna at the satellite's position and setting up the necessary equipment for the support. In order to resolve any conflict, the MCC, RTS or both must be contacted via telephone, given an explanation of the conflict and presented possible alternatives. Depending on the situation, either the MCC will adjust its support requirements or the RTS will accept a shorter turnaround time. In a worst case scenario, the MCC may have this support canceled. However, in all cases, the scheduler does not resolve the conflict without contacting either the MCC or RTS personnel. Once all conflicts are resolved, a conflict free 24 hour schedule is printed and given to the MCCs. A conflict free 24 hour schedule is required before the real time scheduling phase begins.

Real Time Scheduling. The last phase of the scheduling process, known as real time scheduling, occurs during the day of the support and deals with any changes after the 24 hour schedule has been published. The primary reasons for these changes are satellite vehicle problems, remote tracking station problems, changing mission requirements, incorrect requests by an MCC, or incorrect processing of a request by the schedulers. According to Francis Wong, Supervisor of Planning, 21 SOPS, up to a third of the scheduled activities can be changed between the publication of the 24 hour schedule and implementation of the schedule (Wong, 18 June 1992). In some instances, a "ripple effect" can be noticed. A change to one support may require numerous additional changes to maintain a conflict free schedule. A formal priority system is established for all satellite malfunctions occurring in real time. The priority system allows schedulers to determine how serious the malfunction is and what satellite supports can be moved to accommodate the malfunctioning satellite's support requirements.

Research Objective

The objective of this research is to develop an automated scheduling algorithm that generates a feasible initial 24 hour schedule. Specifically, a scheduling algorithm will be developed with the goal of maximizing the number of conflict free supports in the schedule. This goal is equivalent to minimizing the number of conflicts in the schedule. This goal was established since operational schedulers have stated that no conflict or group of conflicts is considered worse than any other conflict. All conflicts are treated as essentially equal since any conflict, no matter how small, requires the scheduler to coordinate with MCC and/or RTS representatives to determine an acceptable solution. Therefore, a schedule that minimizes the number of conflicts will, in general, require less work to resolve the conflicts.

Assumptions. Four key assumptions were required to further define the research effort. These assumptions were developed after discussions with satellite range schedulers and personnel familiar with MCC operations.

The first assumption is that the time window listed for the visibility of a specific satellite by a specific remote tracking station is a hard constraint. Therefore, no attempts will be made to schedule a support outside its time window. In actual scheduling operations, satellites are occasionally scheduled outside their window after coordination with MCC/RTS personnel. Additionally, the turnaround time required to reconfigure for the next support will not be altered from its stated value. Again, in the actual scheduling process, turnaround times are sometimes adjusted. but this is accomplished only after consultation with the MCC and the RTS representatives.

The second assumption is that the candidate remote tracking stations for each satellite support have the necessary equipment to successfully complete the requirements of the support. For example, each remote tracking station will have the equipment to perform actual commanding, telemetry reception, tracking, or other support requirements.

The third assumption is that special case scheduling requirements, such as a required time interval between supports, will not be incorporated into the basic model since these requirements do not occur on a regular basis. Incorporating special request supports into the scheduling program could be an area for further research and is discussed in the conclusions and recommendations in Chapter VI.

The final assumption is that the development of a schedule will be treated as a static problem. A static problem is one in which the schedule is generated only once and no uncertainty exists. Therefore, the algorithm to generate the initial 24 hour schedule will be initiated only after all inputs are received. The schedule will not be built iteratively, as it currently is. Additionally, all time windows, support times, and RTS availabilities are assumed known in advance. The development of the ASTRO system has increased the feasibility of this assumption. Previously, the schedulers iteratively updated the schedule when new support requests arrived. With the advent of ASTRO, the schedulers can (ideally) wait until all requests have arrived, run the automated scheduling program and have the schedule displayed on the ASTRO terminal within minutes.

Document Layout

The rest of this document is structured in the following order. Chapter II examines results found in the literature on previous methods to solve the SRS problem as well as other related solution techniques. Chapter III discusses the problem formulation for an optimal approach to solving the problem, while Chapter IV describes a heuristic solution approach to the problem. Chapter V discusses the results of testing the algorithm using real data sets. Finally, Chapter VI presents conclusions and recommendations.

II. Literature Review

Overview

This chapter surveys the current information that exists about the satellite range scheduling problem. This includes any previous studies in this area or pertinent solution methodologies. The chapter first presents an introduction to scheduling theory and typical goals of common scheduling problems. Next, previous attempts to solve the SRS problem are examined. Finally, any other results reported in the literature which are applicable to the SRS problem are discussed.

Scheduling Theory

In general, scheduling problems are concerned with finding the sequence and timing of activities which are 1) compatible with all given constraints and 2) optimal with respect to some criterion of performance (French,1982:5). A schedule consists of activities (jobs) which must be processed by machines in a specified order at a specified time. In SRS, the activities are the satellite supports and the machines are the remote tracking stations.

Constraints. The sequence of activities that are scheduled often depends upon the three primary types of constraints: technological, precedence, and resource (French,1982:5,48,197). Technological constraints demand that each job be processed through the machines in a specific order (French,1982:5). For example, if the activity is fixing a flat tire, the operation of removing the lug nuts must be performed prior to removing the flat tire. Technological constraints do not apply if the level of scheduling detail in a given problem assumes that activities consist of a single operation. Therefore, technological constraints do not apply in this study since satellite supports are not scheduled to multiple machines (RTSs) in a specific order.

Precedence constraints are similar to technological constraints and limit the schedule flexibility by requiring that certain subsets of jobs be processed in a given order (French,1982:48). A hypothetical example of this would be if at an Air Force maintenance complex all C-141s were repaired before any C-5s were repaired. In the SRS problem, precedence constraints could apply if a satellite has a very important support that requires additional supports every "X" hours to insure that the first support was successful. An example of this would be a satellite support that commands a satellite to discharge and then recharge its battery pack. After this initial important support, additional supports are scheduled every two hours (for example) to monitor the battery pack.

Resource constraints occur when the availability of one or more resources is limited (French,1982:197). In many cases, these constraints are the most limiting ones. Scheduling problems where resources are the limiting factor are known as resource constrained scheduling problems. Resource constraints can be observed in SRS when only a few RTSs are visible to a satellite for a support. The number of RTSs visible to a satellite (the availability of the RTS resource) is a major factor in determining which satellite supports are scheduled to a specific RTS. Generally, more satellite supports are requested than the RTSs can fulfill. Thus, the RTS resource availability is a limiting factor in the number of supports that can be scheduled.

Scheduling Performance Measures. Given that a feasible schedule is possible, the goal of the scheduler is to determine an optimal schedule based on a specific measure of effectiveness (MOE). How a schedule is generated is dependent on the criterion of performance for that schedule. The most common scheduling goal is to minimize the time required to complete all activities (French,1982:12). However, for the SRS problem, a fixed duration of 24 hours exists; therefore, the goal of minimizing the time to complete all SRS activities is not appropriate. Other common measures of performance are to minimize the number of late jobs, minimize machine idle time, or minimize in-process inventory costs (French,1982:13). For most large scale scheduling problems, identifying the measure of performance is of critical importance. For SRS, the goal is to maximize the number of supports scheduled in a 24 hour time period.

Scheduling Conclusion. For most complex scheduling problems, no general computationally efficient solution techniques exist. In essence, scheduling problems address two decision problems: 1) allocation of resources and 2) sequencing decisions. Solving these problems answers two basic scheduling questions: 1) which resources will be allocated to perform each task and 2) when will each task be performed (Baker,1974:5). For the SRS problem, the schedule generated should answer the basic questions of what RTS each support is scheduled at and during what time period in the day will the support occur. Finally, the selection of the appropriate scheduling technique depends on the complexity of the problem, the nature of the model, and the choice of criterion, as well as other factors (Baker,1974:6).

IBM Study on SRS

IBM conducted the only major study to determine the feasibility of computer generated satellite range scheduling during the 1981-1984 time period. The study examined the feasibility of automating three of the four phases of the SRS scheduling process (7-day, 24-hour and realtime scheduling). In an extensive literature review on other attempts to automate SRS, IBM found previous studies examined only very small problems (Arbabi,1984:57). IBM concluded large scale automation of range scheduling was feasible and took a two pronged approach in developing its Continuous Time Scheduling (CTS) algorithm.

CTS Algorithm Development. The two prongs of IBM's study culminating in CTS were to examine mathematical programming approaches and existing heuristic approaches. The mathematical programming approaches examined transportation problems, mixed integer programming (MIP) approaches, and other optimization techniques (Arbabi,1984:59). A mixed integer programming approach was formulated, and it was determined this approach was feasible for up to 50 requests, while storage and run time requirements were unacceptable for more than 50 requests (Arbabi,1984:59). The second prong consisted of developing several heuristic models that attempted to "duplicate" the scheduling rules utilized by the SRS schedulers (Arbabi,1984:59). The steps of the CTS algorithm were not presented; only its results were available. Therefore, the solution methodology of CTS is not exactly known, but its stated capabilities are. Dr. Arbabi stated during a recent telephone conversation that the specific details of the CTS methodology were not releasable (Arbabi, 24 July ,1992).

IBM MIP Formulation. IBM initially attempted to solve the SRS problem using an MIP approach. However, as mentioned above, this approach was not feasible for full scale SRS problems. The IBM MIP formulation is shown below and will be compared to the MIP formulation developed in Chapter III of this research effort.

Decision Variables & Input Parameters. The decision variables and input parameters are defined below. **Note:** In the following, a request specifies the support requirement and a segment is a time window at an RTS in which the request can be satisfied:

i = Request Index

j = Segment Index

k = Segment Index

A_k = Beginning of segment k

B_k = End of Segment k

C_k= Length of request on segment k

$$\delta_{jk} = \begin{cases} 1 & if\ request\ on\ \mathrm{k}\ started\ before\ request\ on\ \mathrm{j} \\ 0 & otherwise \end{cases}$$

M = large positive constant

P = set of pairwise combinations of overlapping segments on each antenna

R_i = Set of segments which service request i

S_k = Offset between the beginning of segment k and the beginning of its request

T_k = turnaround time of the antenna on Segment k

V_k = Preference value for scheduling a request on Segment k.

$$X_k = \begin{cases} 1 & if\ request\ \text{i}\ is\ scheduled\ at\ Segment\ \text{k} \\ 0 & otherwise \end{cases}$$

Note: a request specifies the support requirement and a segment is a time window at an RTS in which the request can be satisfied.

Formulation

Objective Function

$$Maximize \quad \sum_k V_k X_k \quad \forall \ k$$

Subject to:

$$\sum_{k \in R_i} X_k \leq 1 \quad \forall i \qquad (2-1)$$

$$A_k + S_k + C_k \leq B_k \quad \forall \ k \qquad (2-2)$$

$$A_j + S_j + C_j \leq A_k + S_k - T_k + M\delta_{jk} + M(2 - X_j - X_k) \ \forall \ j,k \ \ j \neq k \qquad (2-3)$$

$$A_k + S_k + C_k \leq A_j + S_j - T_j + M(1 - \delta_{jk}) + M(2 - X_j - X_k) \ \forall \ j,k \ j \neq k \quad (2-4)$$

$$S_k \geq 0 \ \forall \ k \quad (2-5$$

$$X_k \ \in \ \{0,1\} \quad (2-6)$$

$$\delta_k \ \in \ \{0,1\} \ \forall (j,k) \ \in \ P \quad (2-7)$$

(Arbabi,1984:60)

IBM Study Conclusions. Dr. Arbabi showed impressive results for the CTS algorithm automating all three sections of the range scheduling process. Ninety-eight percent of all requests were scheduled in less than three minutes (Arbabi,1984:62). However, the Air Force never implemented IBM's range scheduling automation algorithm. The reason, according to Francis Wong, Supervisory Planner, Space Operations Flight, 21 SOPS, was "The proposed IBM system did not offer sufficient flexibility with regard to assigning priorities, accounting for specific satellite and scheduling restrictions, and accomplishing scheduling requirements within defined time constraints." (Wong, 29 January 1993)

Vehicle Scheduling Approach to SRS

Solution techniques for vehicle scheduling problems were examined for their possible application to the SRS problem. This section concentrates on an examination of the general vehicle scheduling problem, similarities between vehicle scheduling problems and the SRS problem, and solution techniques for vehicle scheduling problems.

Vehicle Scheduling Overview. The general vehicle scheduling problem involves scheduling a fleet of vehicles to perform tasks of pickup/delivery for various customers while trying to minimize some measure of performance (such as cost or distance). The scheduler must decide what tasks should be assigned to each vehicle and in what order the tasks should be completed. Three real world constraints commonly determine the complexity of a vehicle scheduling problems . The restrictions are 1) constraints on the length of time a vehicle may be in-service, 2) restrictions that certain tasks must be serviced by specific vehicle types and 3) presence of a variety of depots where vehicles may be housed (Bodin,1983:119). The servicing of tasks within specified time windows is an additional constraint that further complicates vehicle scheduling problems (Bodin,1983:149). Any one of these four constraints can make large scale vehicle scheduling problems challenging to solve.

Vehicle Scheduling Problems and SRS Similarities. There are several similarities between SRS and vehicle scheduling problems. The two primary similarities are: 1) time window constraints, and 2) vehicle restriction types. The time window constraint in vehicle scheduling problems is similar to the SRS requirement that satellite supports must occur within specified time windows. Additionally, the assigning of a restrictive support to a specific RTS is similar to the vehicle restriction constraint where a specific vehicle must deliver a certain package. In SRS, for certain classes of satellites, only a few RTSs will be visible to a satellite during a pass. For example, a low altitude satellite may be visible to only one RTS and the satellite must be scheduled for its support at that RTS. These similarities make solution methodologies of some specific vehicle scheduling problems applicable to the SRS problem.

Vehicle Scheduling Problem Solution Methodologies. Vehicle scheduling problems are usually solved one of two ways: 1) a network flow approach (Bodin,1983:129) or 2) a heuristic approach (Solomon,1987:254). For vehicle scheduling problems with time window constraints, the network flow approach is generally not applicable. Therefore, only the heuristic approaches will be examined in more detail.

Heuristic Approaches. For the vehicle routing and scheduling problem with time windows (VRSPTW), heuristic approaches offer the most promise for practical size problems (Solomon,1987:254). Vehicle scheduling problems with time window constraints occur frequently with business organizations that work on fixed time schedules. Most of the algorithms developed for VRSPTW are extensions of vehicle routing problem algorithms that include the time element in the algorithm methodology (Solomon,1987:255). In his article, Solomon compared the performance of several heuristics under varied conditions to determine the best performing heuristic.

The primary heuristics Solomon compared were insertion heuristics. An insertion heuristic sequentially builds a schedule by inserting an unscheduled customer into the schedule based on some measure of performance. An insertion heuristic will insert one customer at a time into the schedule by searching the feasible locations within the schedule and selecting the location that optimizes the measure of performance. This process is repeated until no additional customers can be scheduled. The measure of performance for insertion can vary from problem to problem, but common measures of performance include: 1) minimize the distance from nearest scheduled customer (nearest neighbor), 2) arbitrarily selecting an insertion location, and 3) the "cheapest" insertion based on a selected cost value (Syslo,1983:362). For VRSPTW problems, a good insertion technique is one that minimizes the amount of time a customer deviates from his/her desired pickup/delivery time. (Solomon,1987:257).

The results of the comparisons of algorithm performance indicate insertion heuristics perform best for most VRSPTW problems. Solomon stated these heuristics perform best for problems with many customers per vehicle, a high density of customers with time windows, and tight time window constrainsts (Solomon,1987:263).

The key characteristics of the SRS problem are 1) many supports per RTS, 2) a very high percentage of supports in time windows and 3) tight support time window constraints. These characteristics are very similar to those mentioned above, which indicates that an insertion approach may be appropriate for the SRS problem.

Computational Complexity

Computational problems are often classified based on their complexity. According to Parker, complexity theory seeks to classify problems in terms of the mathematical order of the computational resources required to solve the problems via digital computer algorithms (Parker and Rardin,1982a:4). The goals of the theory are to broadly classify problems and algorithms according to their convenience for solution (Parker and Rardin,1982a:3). In complexity theory, there are two broad classes of decision problems: 1) polynomial *P* and 2) nondeterministic polynomial *NP*. In order to qualify for the Class *P*, a decision problem must actually be solvable in polynomially bounded time (Parker and Rardin,1982a:8). The class NP includes all decision problems for which the correctness of a solution can be verified within a polynomial amount of time. The hardest problem in NP are classified as NP-complete. The general resource constrained scheduling (RCS) problem is in this class. The only known solution methods for NP-complete problems can take an inordinate amount of time on the fastest computer in existence if the size of the problem is big enough. The SRS problem is a type of resource constrained scheduling problem and we do not know of a way to solve it that is significantly faster than for the more general RCS problem. However, as French notes the classification of a problem as NP-complete is not sufficient reason to resort to heuristic methods. The problem must also be large enough that enumerative methods are computationally impractical (French,1982:156). This lends credence to the approach of first investigating optimal approaches to solving the SRS problem.

III. Mixed Integer Programming Approach to the SRS Problem

Model Formulation

The focus of this chapter is the formulation and solution of the Satellite Range Scheduling (SRS) problem using mixed integer programming (MIP). The MIP approach is utilized in many scheduling applications as a modeling technique, especially when a possible scheduling assignment can be represented by a binary decision variable. For the SRS problem, the decision variables in the MIP represent whether or not a satellite support is scheduled at a remote tracking station (binary decision variable), and, if the support is scheduled, the start time of this support (continuous decision variable).

Several factors influence the number of supports that can be scheduled. The altitude of a satellite (low, medium, or high), number of candidate remote tracking stations to which a support can be scheduled, and the support length can greatly affect whether or not the satellite support can be scheduled. According to Ken Chambers, technical advisor for Space Operations Flight, 21 SOPS, approximately 40% of the supports are low altitude supports, 20% are medium altitude supports, and 40% of the supports are high altitude supports (Chambers, 24 October 1992). For a low altitude support, there is little scheduling flexibility as the start time of the support is the same as the beginning of the visibility window and the support requirement ends at the end of the visibility window. For medium and high altitude satellite supports, there is greater flexibility in the start time of the support due to the longer visibility windows. Another key factor in determining the number of scheduled supports is the support length. The total support length is comprised of the length of the support requirement and the turnaround time for the RTS where the support is scheduled. The turnaround time is the amount of time required by a RTS to reconfigure for a support. The length of the support requirement depends on type of satellite (low, medium, or high altitude) and the mission of the satellite.

The turnaround time at a RTS depends primarily on the type of satellite. For a low altitude satellite, the turnaround time is normally twenty minutes, while the turnaround time at a RTS for a medium/high altitude satellite is fifteen minutes. All of the above factors influence the number and type of satellite supports that can be scheduled.

The objective of the MIP model is to maximize the number of supports scheduled. The key constraints limiting the number of scheduled supports are the small number of RTSs, each support must be scheduled within its visibility window, and no overlapping supports may be scheduled. An overlapping support is defined as two supports assigned to the same RTS which are scheduled with some portion of their support time windows intersecting. In the remainder of this chapter, the input parameters, decision variables, objective function, and constraints of the MIP model will be examined in detail. Finally, the analysis on the size of the MIP model and the impact of these findings on the solution methodology will be presented.

Input Parameters

Input parameters are known constants that affect the value of the decision variables. The index i refers to a specific support, while index j denotes a specific RTS. Additionally, m denotes the number of RTS antennas (16) and n denotes the number of supports requested for a given day.

Using the above indices, the input parameters for the MIP problem formulation are as follows:

BV_{ij} - Beginning of visibility window for support i at RTS j

EV_{ij} - End of visibility window for support i at RTS j

R_{ij} - Length of required support i at RTS j

TO_i - Turnaround time for support i

SDT_j - Start of downtime for RTS j

EDT_j - End of downtime for RTS j

RTS_i - represents the set of RTSs that support i is visible to M - large positive constant value

Note that we are assuming there is no more than one scheduled downtime at an RTS per day. This can be easily modified to allow multiple downtimes.

Decision Variables.

There are three sets of decision variables in the SRS formulation. The first decision variable, X_{ij}, is a binary variable that is set to one when support i is scheduled at RTS antenna j and zero otherwise. The range of subscript j is restricted to RTS_i.

The second set of decision variables designate the start time of support i at RTS antenna j, denoted by ST_{ij}. Two factors affect the value of ST_{ij}: 1) the visibility window of support i, and 2) the availability of RTS antenna j. ST_{ij} is the only continuous decision variable and is required to be nonnegative.

The last decision variable is used to ensure overlapping supports are not scheduled. The binary variable y_{ihj} denotes the relative order of the start times of supports i and h at a RTS j. Notationally, y_{ihj} is defined as:

$$y_{ihj} = \begin{cases} 1 \quad if \quad \mathrm{ST}_{hj} \; < \; ST_{ij} \quad i \; \neq \; h \\ 0 \quad if \quad \mathrm{ST}_{hj} \; \geq \; ST_{ij} \end{cases}$$

SRS MIP Formulation

Now that all the indices, input parameters, and decision variables have been defined, the objective function and constraint sets for the SRS MIP formulation can be discussed. The complete formulation is shown below.

Objective Function:

$$Maximize \sum_{i=1}^{n} \sum_{j \in RTS_i} X_{ij} \qquad (3-1)$$

Subject to:

$$\sum_{j \in RTS_i} X_{ij} \leq 1 \quad i = 1 \ \cdots \ n \qquad (3-2)$$

$$ST_{ij} \geq BV_{ij} * X_{ij} \quad i = 1 \cdots n, \ \forall j \in RTS_i \qquad (3-3)$$

$$ST_{ij} \leq (EV_{ij} - R_{ij}) * X_{ij} \quad i = 1 \cdots n, \ \forall j \in RTS_i \qquad (3-4)$$

$$STij + R_{ij} \leq SDTj + M(1 - X_{ij}) \quad \forall i \ given \ j \qquad (3-5)$$

$$ST_{ij} \geq EDTj + TO_i - M(1 - X_{ij}) \quad \forall i \ given \ j \qquad (3-6)$$

$$ST_{ij}+R_{ij}+TO_h \leq ST_{hj}+My_{ihj}+M(1-X_{ij})+M(1-X_{hj}) \ \forall \ h \, \forall i \ given \, j \qquad (3-7)$$

$$ST_{hj}+R_{hj}+TO_i \leq ST_{ij}+My_{ihj}+M(1-X_{ij})+M(1-X_{hj}) \;\forall\; h \,\forall\, i \; given \; j \quad (3-8)$$

$$\begin{aligned} X_{ij} &\in \{0,1\} \\ y_{ihj} &\in \{0,1\} \\ ST_{ij} &\geq 0 \,\forall\, i,\; j \in RTS_i \end{aligned}$$

Objective Function

$$Maximize \sum_{i=1}^{n} \sum_{j \in RTS_i} X_{ij} \qquad (3-1)$$

The objective function for this problem is to maximize the number of supports scheduled.

Constraints

The constraints for this formulation are:

- schedule each support at most once
- schedule each support in its visibility window
- schedule no supports during RTS downtime
- schedule no overlapping supports

In each constraint set, only values of j contained in the set RTS_i are considered. Since both the total number of supports and the number of candidate RTS antennas for a support vary from day to day, the number of individual constraints for each constraint set varies for each day scheduled.

Schedule each support at most once. This constraint insures that a support is not scheduled to more than one RTS antenna. Also, each support i has a candidate list of RTS antennas the support can be scheduled at (labeled RTS_i).

$$\sum_{j \in RTS_i} X_{ij} \leq 1 \quad \forall\, i \qquad (3-2)$$

The total number of constraints for this constraint set may be estimated by multiplying the total number of supports by the average number of RTSs visible to a satellite.

Schedule each support in its visibility window. Each satellite has a time window for each RTS antenna it is visible to. If the satellite is scheduled at that RTS the satellite support must occur within this time window. The support must start after the satellite is visible to the RTS and end before the satellite is no longer visible to the RTS.

$$ST_{ij} \geq BV_{ij} * X_{ij} \quad i = 1 \cdots n, \;\; \forall\, j \in RTS_i \qquad (3-3)$$

$$ST_{ij} \leq (EV_{ij} - R_{ij}) * X_{ij} \quad i = 1 \cdots n, \;\; \forall\, j \in RTS_i \qquad (3-4)$$

The total number of constraints in this constraint set may be estimated by multiplying the number of supports by the average number of RTSs visible to a support times two.

Schedule no supports during RTS downtime. When a remote tracking station is down for repair, satellite supports cannot be scheduled to that remote tracking station. The number of supports that are affected by this constraint set depends on the number of satellites visible to the RTS and the length of the downtime.

$$STij \; + \; R_{ij} \leq SDTj + M(1 - X_{ij}) \quad \forall \, i \;\; given \;\; j \qquad (3-5)$$

$$ST_{ij} \geq EDTj \; + \; TO_i \; - \; M(1 - X_{ij}) \quad \forall \, i \;\; given \;\; j \qquad (3-6)$$

This constraint set can be avoided for low altitude satellite supports by preprocessing each problem's input. When the downtime for each RTS is known, all supports that are visible to that RTS can be examined and any support whose visibility window overlaps with this downtime window cannot be scheduled at that RTS. For example, if support 5 is visible to RTS LION-A from 0820-0835, but LION-A is down from 0500 to 0830, then the LION-A is no longer considered as a possible RTS for support 5 to be scheduled at. This check is accomplished when the data is preprocessed and input to the MIP solver.

Schedule no overlapping supports. This constraint set insures that two supports scheduled at the same RTS antenna do not overlap.

Constraints sets 3-7 and 3-8 ensure that no overlap exists between support i and any other scheduled support (labeled support h) at RTS j. Constraint set 3-7 ensures that that no overlap exists between support i and support h when support

h is scheduled scheduled after support i. In this case, the start time of support h must be greater than the start time of support i plus support i's length plus the turnaround time for support h. Constraint set 3-8 ensures that that no overlap exists between support i and support h when support h is scheduled before support i. In this constraint set, the start time of support i must be greater than the start time of support h plus support h's length plus the turnaround time for support i. For either constraint 3-7 or 3-8 to be enforced, both support i and support h must be scheduled at RTS j. If either of these supports is not scheduled at RTS j (i.e. $X_{ij} = 0$ or $X_{hj} = 0$), then a large value M is added to the right-hand-side of the constraint and this relaxes the non-overlapping restriction. Additionally, constraint 3-7 will only be enforced when support h follows support i on RTS j ($y_{ihj} = 0$). Constraint 3-8 will only be enforced when support i follows support h on RTS j ($y_{ihj} = 1$).

$$ST_{ij}+R_{ij}+TO_h \leq ST_{hj}+My_{ihj}+M(1-X_{ij})+M(1-X_{hj}) \ \forall \ h \, \forall \, i \; given \; j \quad (3-7)$$

$$ST_{hj}+R_{hj}+TO_i \leq ST_{ij}+My_{ihj}+M(1-X_{ij})+M(1-X_{hj}) \ \forall \ h \, \forall \, i \; given \; j \quad (3-8)$$

Comparison with IBM MIP Formulation

IBM, as part of its efforts to automate the SRS process, developed their own MIP formulation of the SRS problem. IBM examined scheduling every support using a MIP approach. The formulation for this research effort was developed independently of the IBM formulation. A comparison has been made to determine similarities and differences between the two formulations. The primary purpose of the comparison was to verify that the formulation developed for this research effort is complete and did not overlook any constraints or restrictions on the SRS problem. The complete IBM formulation was shown in Chapter II and will be referenced when comparing the two formulations.

Overall, the two formulations are very similar. The objective function and general constraint types are the same for each formulation except the MIP formulation in this research effort includes constraints to ensure that no supports are scheduled during a major RTS downtime. The primary differences in the formulation is how a start time of a support is identified. The formulation for this research identifies specific start times for each scheduled support. The IBM formulation identifies an offset between the beginning of a time segment and the beginning of a support request. Therefore, the start time of a support in the IBM formulation is identified by adding (subtracting) the offset to the beginning (end) of a segment.

Number of Variables

The number of integer variables is a key factor in determining if the problem can be modeled as an by integer programming problem and solved in a reasonable amount of time . The exact number of integer variables for a given problem, the expected number of integer variables, and and the impact of the number of integer variables will now be discussed.

Number of Variables for a Specific Problem. The number of integer variables needed to model SRS problems differs with each problem. For a given problem, the number is the sum of the number of allowable support and RTS combinations (labeled N_{ij}) and the total number of possible overlapping supports, (labeled OV_{ihj}). OV_{ihj} is set to one when support i and support h are both visible to RTS j and their visibility windows overlap. For problem sets with only low altitude supports, determining the number of overlapping supports is a straightforward task of comparing the beginning and ending visibility values for supports visible to the same RTS. For problem sets involving low, medium, and high altitude supports, the number of overlapping supports rapidly increases. The two primary reasons for the increase are 1) the long visibility windows of high altitude satellite supports and 2) the large number of RTS antennas a high altitude satellite support is visible to. A single high altitude support that is visible to several RTSs for the entire day will overlap with every other support at each RTS the support is visible to. Mathematically, the total number of integer variables is:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} N_{ij} + \sum_{i=1}^{n}\sum_{h=1}^{n}\sum_{j=1}^{m} OV_{ihj} \quad \forall\ i, feasible\ h, \quad \forall\ j \in RTS_i\ and\ RTS_h$$

Expected Number of Integer Variables. The expected number of integer variables provides an estimate of the actual number of variables needed to model a specific problem. The expected number is a function of the number of feasible support/RTS combinations and the number of overlapping supports. For problems involving low, medium and high altitude supports, the number of integer variables would be very large. The number of integer variables in the problem would equal:

$$(\sum_{i=1}^{n}\sum_{j\in RTS_i} N_{ij}) + ((HAS + LAS) + HAS + LAS)$$

where:

$\sum_{i=1}^{n}\sum_{j\in RTS_i} N_{ij}$ = total number of feasible support/RTS combinations.

HAS = the number of medium/high altitude supports

LAS = the number of low altitude supports.

For example if there were 200 high altitude satellite supports, 150 low altitude satellite supports, and 2150 possible support/RTS combinations, the total number of integer variables would be $2150 + 200 * 150 + 200^2 + 150 = 72,300$ integer variables. However, according to Ken Chambers, for problems involving low altitude supports only, the expected number of variables would be approximately three times the total number of low altitude supports This is due to the fact that a low altitude satellite support is visible to an average of one and one-half feasible RTS antennas per support and the short visibility windows reduce the number of overlapping supports (Chambers, 15 January, 1993).

Impact of Number of Integer Variables. The computational complexity of integer programming problems limits the size of problems which can be modeled and solved in a reasonable amount of time. The size of the problem that can be modeled and solved is directly related to the number of integer variables. The MIF approach can take an amount of time which is an exponential function of the the size of the problem (number of integer variables). As a result, trying to model and solve the entire SRS problem as a MIP is not considered a feasible approach.

Scheduling Low Altitude Satellite Supports by MIP

The approach developed in this research effort schedules only the low altitude satellites supports using the MIP formulation. The primary reason this approach was taken is the fact that the number of integer variables required to model this problem is significantly reduced when only those satellite supports are scheduled. The number of integer variables when just low altitude satellite supports are scheduled is approximately three times the total number of low altitude supports. The number of X_{ij} variables is decreased since these satellite supports are only visible to a few RTSs. The number of y_{ihj} variables is decreased since a lower number of satellite supports overlap due to the short visibility windows of low altitude satellite supports. In fact, in many cases a low altitude support has only one candidate RTS. This reduction in the number of integer variables enables a larger number of satellite supports to be scheduled using a single MIP formulation. Several test cases were run and up to eighty-five low altitude satellite supports have been scheduled in a **timely** manner when solving the MIP. A timely manner is defined as MIP solution times of ten minutes or less. On most days, there are normally between one hundred forty and one hundred seventy low altitude supports that need to be scheduled during a given day. Therefore, more than one MIP is required to schedule all the low altitude supports for a single day. How these solutions are linked together and how the MIP solution technique fits into the overall approach for scheduling satellite supports will be discussed in Chapter IV.

IV. SRS Heuristic Approach

Overview

In this chapter, the heuristic approach used to develop the initial 24 hour schedule will be presented. The heuristic consists of two sets of algorithms: 1) schedule generation algorithms and 2) schedule improvement algorithms. The heuristic approach is a two phased approach: phase one schedules as many low altitude satellite supports as possible, while phase two schedules as many medium/high altitude satellite supports as possible. In each phase, a schedule generation algorithm initially schedules as many satellite supports as possible. Then a schedule improvement procedure attempts to adjust the schedule so that unscheduled satellite supports can be added to the schedule. The schedule generation algorithms are the MIP linking procedure and a satellite insertion procedure. The MIP linking procedure schedules the low altitude satellite supports. The satellite insertion procedure schedules the medium/high altitude satellite supports. The schedule improvement algorithms are a two satellite interchange procedure and a three satellite interchange procedure. Figure 4.1 shows the flow of the SRS heuristic approach.

The topics in this chapter will be covered in the following order.

- schedule generation procedures
- general schedule improvement methodology
- the specific implementation of the interchange procedures

Schedule Generation Algorithm for Low Altitude Satellite Supports

The low altitude satellite supports were scheduled using the MIP approach developed in Chapter III. However, a linking procedure was required to schedule all the low altitude supports for a single day since the number of low altitude satellite

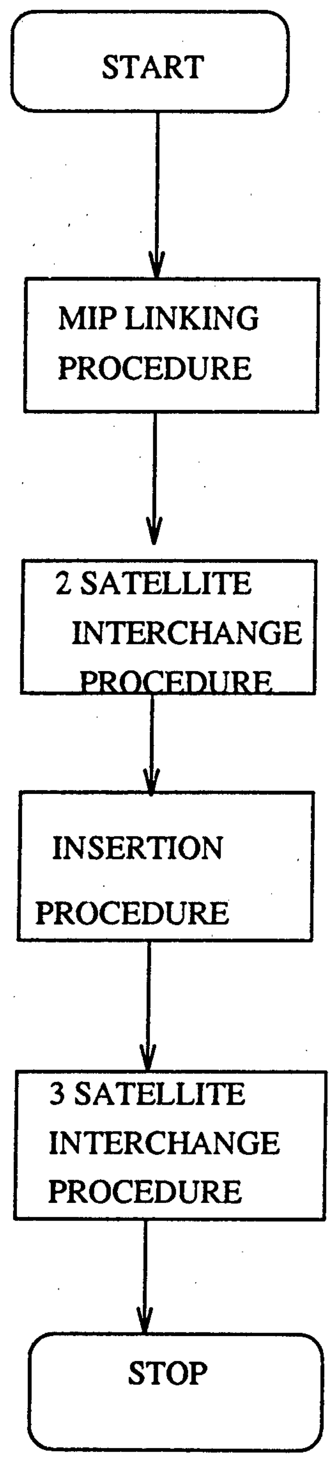

Figure 4.1. SRS Heuristic

supports for an entire day is greater than the number of supports that can be scheduled in a timely manner using a single MIP. In general terms, the linking procedure divides the day into two blocks, schedules the low altitude satellite supports in the first block, and then incorporates the results of the first block when scheduling the low altitude satellite supports in the second block. This section will examine the three key steps in the linking procedure: 1) preprocessing the data, 2) solving the MIP SRS model, and 3) incorporating the results of the previous block in the data for the next block. Figure 4.2 shows the flow of the MIP linking procedure.

SRS Preprocessing. The first step in the linking procedure is preparing the data so it can be used to formulate the MIP

Preprocessing of SRS Data. The SRS data used was the ASTRO general list database provided by Space Planning Flight, 21 SOPS, which lists all the information for each support. Two primary steps were required before the data was in the proper form for the input tables: 1) separate the SRS database into two databases with one for low altitude satellite supports and the other for medium/high satellite supports, and 2) sort the low altitude satellite support database. Appendix B shows the format of each database.

Step 1 The first step in the preprocessing of the SRS data was to separate the low altitude satellite supports and medium and high altitude satellite supports into separate databases. The visibility window for each support was used to distinguish the low altitude supports from the medium/high altitude supports. All supports with a visibility window of thirty minutes or less were categorized as low altitude supports while supports with visibility windows of greater than thirty minutes were categorized as medium/high altitude supports.

Step 2 The second step in preprocessing the SRS data was to sort the database based on BV_{ij}. The first entry in the sorted database would be the support and specific RTS combination with the earliest visibility, the second entry would be the

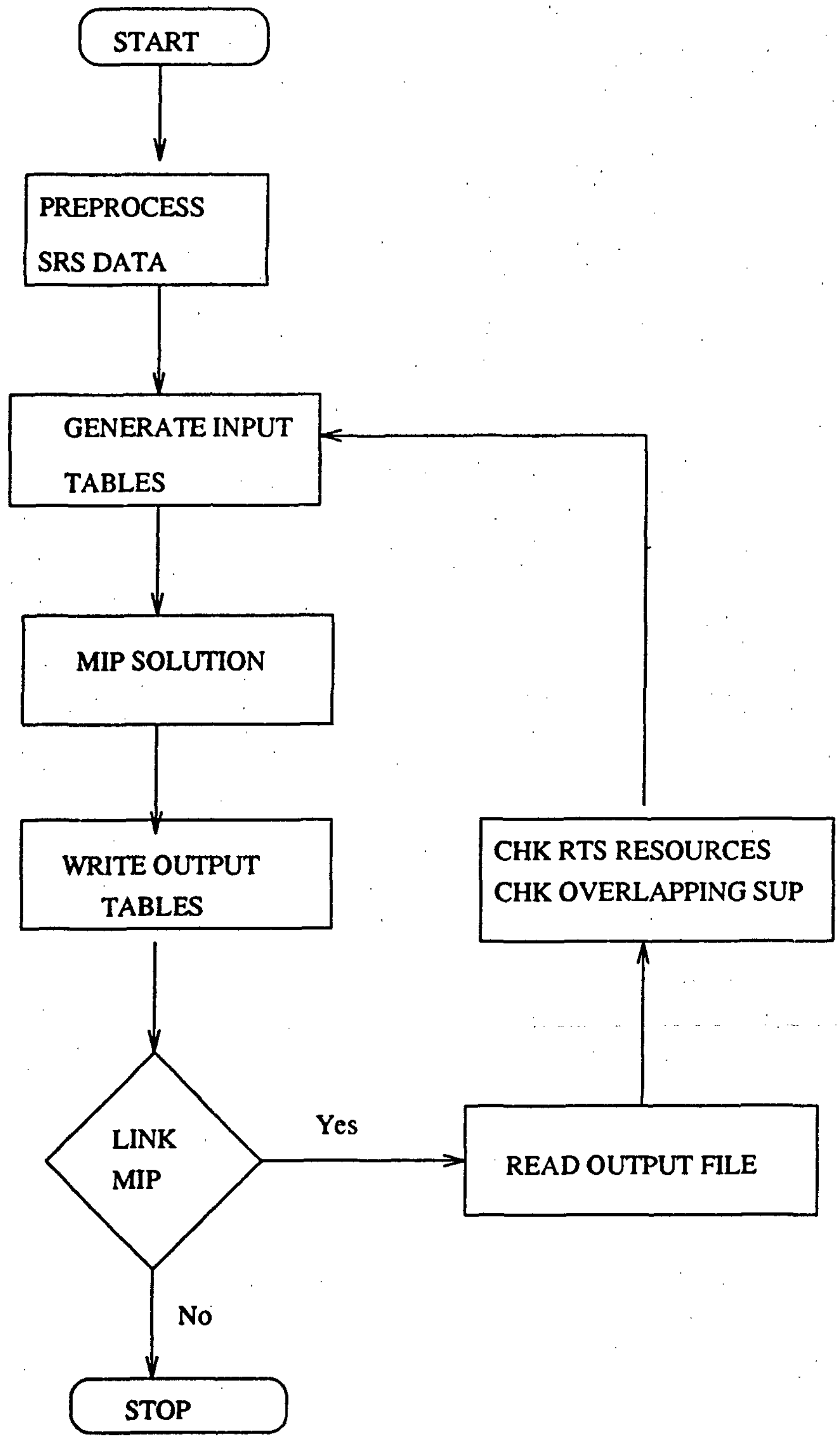

Figure 4.2. Flow of MIP Linking Procedure

support/RTS combination with the second earliest beginning of visibility and so on. Entries for a single support could be scattered in the database since the beginning of visibility time for a support at different RTSs varies by as much as one-half hour.

MIP Input Tables. Once the SRS data was preprocessed, the necessary input tables for the MIP model were then generated using the data frcm the sorted database. The requested supports are divided up into problems with no more than eighty- five supports. Therefore, the data for blocks of eighty-five supports is included in each set of input tables.

MIP Model Solution. The MIP model associated with a block of satellite supports is generated using the input tables. Upon solution of the model, an output file is created which shows what supports are scheduled to each RTS and the start times of each of the scheduled supports. If additional unprocessed blocks remain, the solution is then read by a program which generates the data needed for the next block of data.

Incorporating Results in Next MIP Block. The results from the solution of the previous block are examined when developing the input tables for the next block. When linking MIP blocks, the procedure must ensure that 1) no support scheduled in a previous block is rescheduled in the next block and 2) the RTS resources used for supports in a previous block are not reallocated for a support at the beginning of the next block. In the first step, a support that is not scheduled in the previous block and has an entry in the next block, is entered in the input data for the next block. Additionally, satellite supports associated with a support scheduled in a previous block's solution are deleted from any subsequent block's MIP input table. In the second step, parameters in subsequent block formulations are initialized so that RTS resources are unavailable until previously scheduled supports are completed. For the test data, only two blocks are required although the described procedure is capable of processing additional blocks.

Schedule Generation Algorithm for Medium/High Altitude Satellite Supports

An insertion procedure was developed to schedule medium/high altitude supports around scheduled low altitude supports. Generally, the insertion procedure determines the order in which the medium/high altitude supports will be inserted into the schedule, the possible RTSs where an insertion can occur, and the best time within a visibility window for the insertion. The best time for insertion is the time that minimizes the amount of free time in the schedule. Free time is defined as an amount of time between the end of the inserted support and the turnaround time of the support scheduled immediately after this support. Figure 4.3 illustrates this concept. In this example, Support D is considered for insertion into the schedule. The insertion algorithm will insert support D after support A since (for illustration purposes) the least amount of free time in the schedule occurs when support D is inserted after support A and before support B. The free time for this insertion is approximately ten minutes as compared with approximately forty five minutes if insertion after support B was performed. How the SRS insertion procedure determines the RTS and time of insertion for each medium/high altitude support is presented next.

The SRS insertion algorithm has three major steps: 1) rank order the medium and high altitude supports by decreasing difficulty of insertion (the most difficult support is first), 2) determine which RTSs the support is visible to and rank order these RTSs by increasing "free time", and 3) find the best insertion time for the support. The "best" time is determined using a combination of a first feasibility selection rule and best insertion time rule. The three major rules are explained below. Figure 4.4 shows the flow of the insertion procedure.

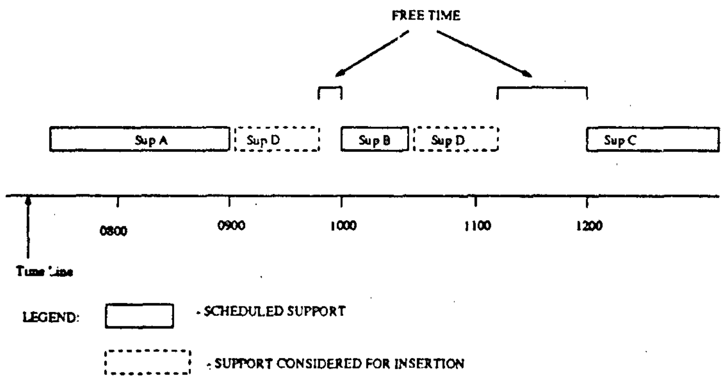

Figure 4.3. Free Time Determination

Rank Order Supports. The first step in the insertion procedure is to rank order the supports by decreasing difficulty of insertion. All the medium and high altitude supports are ranked based on two criteria: 1) the ratio of support requirement length (not including turnaround time) to the support's request window (the larger the ratio, the more difficult the support is to schedule), and 2) flexibility of the support (number of RTSs visible to the support). The request window is the time window the support should be scheduled in and is determined by MCC personnel. In most cases, the request window for a support is considerably smaller than its visibility window. The ratio of support length to request window length is considered a good measure since, in general, the difficulty of scheduling a support must take into account both the length of the support and the length of the request window. For example, a support with a fifteen minute support length and a forty five minute request window (ratio 15/45) is more difficult to schedule than a support with a thirty minute support length and an two hour request window (ratio 30/120). For supports that have the same support length to visibility window length ratio, the support that is visible to fewer RTSs would be ranked higher. The goal is to insert the more difficult supports first since there is more free time available in the schedule.

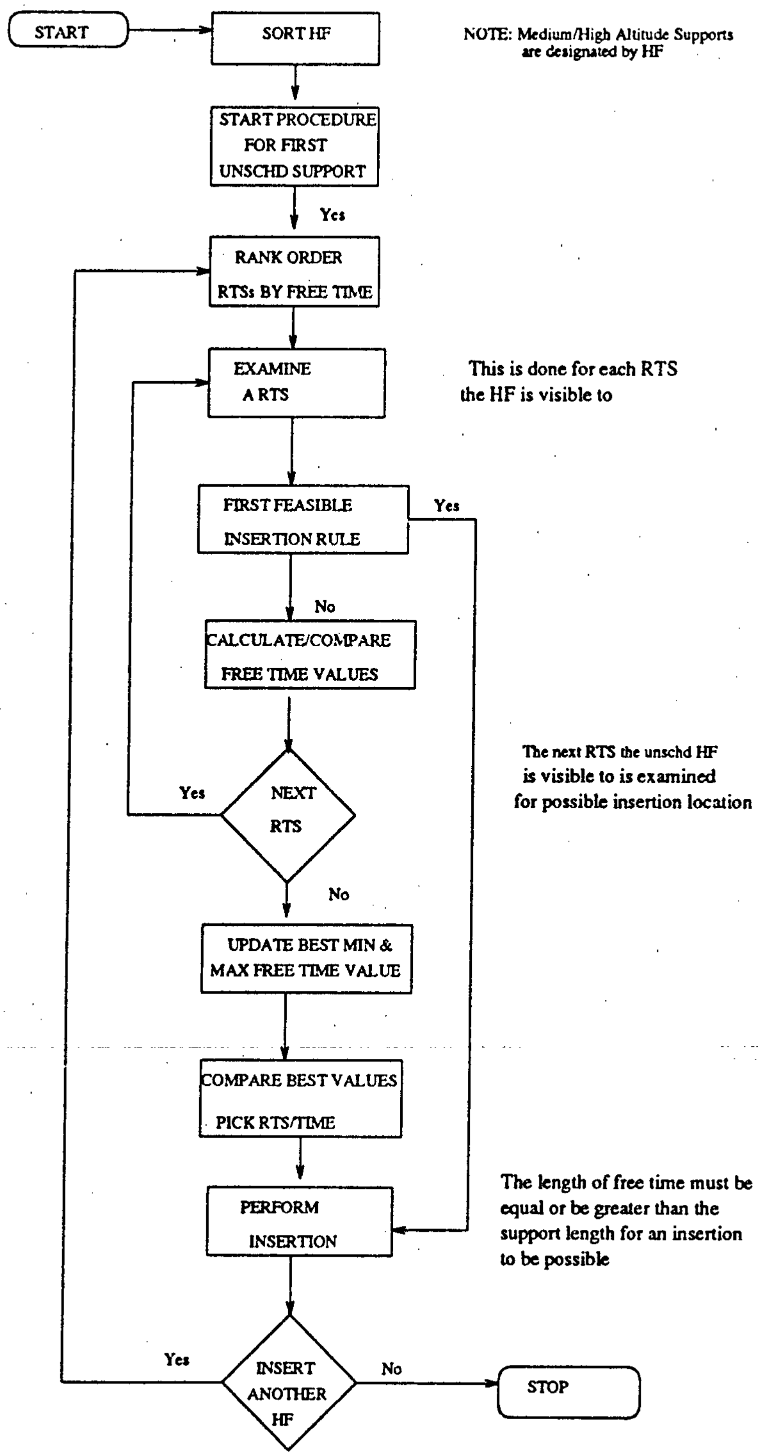

Figure 4.4. Flow of SRS Insertion Procedure

Determine & Rank Order RTSs. For each support in the list, the RTSs the support is visible to are identified and then rank ordered based on the free time of each RTS. Therefore, the RTS with the most free time is considered first for possible insertion. Free time for a RTS is defined as the total time available during a day minus the total time required for scheduled activities (all scheduled support lengths, turnaround times and downtime). Free time is an approximate measure of how much of the total time is available at a RTS for additional supports to be scheduled. The measure is approximate since free time does not take into account the amount of free time inside the support's visibility window or unusable time at that RTS. Unusable time is free time between scheduled supports that is not long enough to accommodate an unscheduled support. An example of unusable time would be ten minutes between the end of one support and the beginning of the next support's turnaround time at a RTS. Finally, after the RTSs are rank ordered, insertion of the unscheduled support is attempted at each RTS until either the best insertion time is determined or no possible insertion time can be found.

Determine Best Insertion Time. The last and most important step in the insertion procedure is to determine at what RTS and time period within the day the unscheduled high or medium altitude support should be scheduled. A preliminary criteria for the insertion procedure is that a block of free time exists within a RTS that is larger than the request length plus turnaround time for the support. If no block of free time exists, the support cannot be inserted into the schedule. The insertion procedure finds an insertion time that minimizes the amount of free time in the schedule.

The criteria for an insertion is a combination of the first feasible insertion and best insertion time rules. These rules are applied consecutively for each attempted support insertion into the schedule and are explained below.

First Feasible Insertion Rule. The first feasible insertion rule has the procedure stop searching for any more insertion times for the current unscheduled support if the current time meets one of two criteria: 1) if after the support is inserted, there is less than five minutes of free time between the inserted support and the next scheduled support or 2) if after the support is inserted, there is more than one hour of free time between the inserted support and the next scheduled support. If both of the stopping criteria are met, the procedure inserts the support in the time location where there is less than five minutes of free time (rule 1). The rationale for these rules was established after talking with people familiar with the range scheduling process and examining the available data. The five minute criteria was chosen since the probability of finding another insertion time leaving less than five minutes of free time is small. The second criteria was arrived at after examining the data available. Approximately, ninety percent of all medium/high altitude supports are less than one hour in total support length (forty five minute support requirement plus fifteen minute turnaround time). Therefore, using this criteria, a large majority of supports can be scheduled in a free one hour period.

Best Insertion Time Rule. If the first feasibility insertion rule is not satisfied, the procedure keeps searching for a time based on the best insertion time rule. This rule is based on minimizing the amount of unusable time within the schedule. Every RTS visible to an unscheduled support is examined and, if possible, two values are computed: a minimum free time value and a maximum free time value. These values are then compared to determine the best insertion time. The steps of the best time rule are presented below.

Step 1 Calculate the "minimum free time" value and the "maximum free time" value over all possible insertion times at a single RTS. The minimum free time value is the lowest amount of free time between the end of the support requirement for the inserted support and the beginning of the turnaround time for the support scheduled immediately after this support, while the maximum free time value is the largest amount of free time between the end of the support requirement for the inserted support and the beginning of the turnaround time of the support scheduled immediately after this support. Every possible insertion time at a RTS is examined before these values are determined.

Step 2 Compare the values calculated in Step 1 with the best minimum and maximum free time values to date (called the incumbent minimum and maximum free time value). If the current minimum free time value is less than the incumbent minimum free time value, then the incumbent minimum free time value is updated. Likewise, if the current maximum free time value is greater than the incumbent maximum free time value, then the incumbent maximum free time value is updated.

Step 3 If the support is visible to another RTS, steps 1 and 2 are repeated. If the support is not visible to another RTS, proceed to step 4.

Step 4 After all the visible RTSs have been checked, the incumbent minimum and maximum free time values arc compared to determine the insertion time. If either value is greater than thirty, then the value has thirty subtracted from it. The reason for this is to leave free time which is usable for later unscheduled supports that have a total support length of less than thirty minutes. The procedure then compares the two values and chooses the smaller one which indicates the RTS and time the support will be inserted at.

Flow of Insertion Procedure. The first feasible insertion and best insertion time rules are consecutively applied to every RTS the support is visible to. If either of the first feasible insertion criteria is met, the insertion procedure adds the support to the schedule and proceeds to the next unscheduled support for insertion. If neither of the first feasibility insertion criteria is met, the minimum and maximum free time values are then calculated. If every RTS the support is visible to has been examined, and the first feasibility insertion criteria have not been met, the incumbent minimum and maximum free time values are then compared to determine the insertion time.

SRS Insertion Example. The following example will illustrate the techniques utilized in the insertion procedure. The example problem will go step by step through each phase of the insertion procedure. Table 4.1 displays the unscheduled supports for the example problem.

Table 4.1. Insertion Example

Sup No	Ident	BV	EV	Req Len	TAT	RTS1	RTS2	RTS3	RTS4
123	2567	20	447	75	15	GUAM-B	HULA-B	INDI-A	BOSS-A
255	4373	16	775	75	15	BOSS-B	LION-A	COOK-B	POGO-A
212	9434	0	1440	60	15	PIKE-A	COOK-B	LION-B	
98	9366	0	1440	60	15	POGO-A	COOK-A	LION-B	PIKE-A

The first item to note about Table 4.1 is the order of the supports. The supports are ordered from the largest ratio to the smallest ratio of requirement length to length of the visibility window. Support 123 is the first candidate support since it has the largest ratio (75/427)and support 255 is the second candidate since it has the second largest ratio. Supports 212 and 98 have the same ratio, but support 212 is the third candidate support for insertion since it is less flexible (visible to only three RTSs, compared to four for support 98). For illustration purposes, we will assume that each of the RTSs for each support are listed in order of free time. Note that after each insertion, the RTS free time values are updated, possibly leading to a reordering of these RTSs. Therefore in this example, the first two steps of the insertion procedure were performed to arrive at Table 4.1.

The insertion procedure begins with support 123. The procedure will attempt to insert this support at RTS GUAM-B first. We will assume the procedure found an unscheduled time segment starting at time 580 with a minimum free time of 15 minutes (this means a block of time of 105 minutes was found $(105-(75+15)=15)$ and an unscheduled time segment starting at time 95 with a maximum free time of 48 minutes. This means that at RTS GUAM-B a free block of 105 minutes was the smallest free block found and a free block of 138 (75 + 15 + 48) minutes was the largest free block found when all possible insertions at this RTS were examined. Since neither of our stopping criteria from the first feasible insertion rule are met, the procedure updates the incumbent minimum and maximum free time values and attempts to insert 123 at RTS HULA-B. For this RTS, a minimum free time of 4 minutes was found at an unscheduled time segment starting at time 305 and a maximum free time of 42 minutes was found at an unscheduled time segment starting at time 234. Since the stopping criteria of free time less than 5 minutes was met, the search for an insertion time is stopped and support 123 is inserted at RTS HULA-B with a start time of 320(305 + 15 = 320). The free time for RTS HULA-B will now be updated. For example, if HULA-B had a free time of 980 minutes before support 123 was inserted, its new free time would be 890 (980 − 90). RTS INDI-A will not be looked at for an insertion time since support 123 has been scheduled.

Now the insertion procedure considers support 255. Again the RTS with the most free time, BOSS-B, will be examined first. For this RTS, the minimum free time of 23 minutes was found at an unscheduled time segment starting at time 145 and a maximum free time of 27 minutes was found at an unscheduled time segment starting at time 224. Since neither of the first feasible insertion criteria is met, the incumbent minimum and maximum free time values are updated and the current schedule of RTS LION-A is searched for an insertion time. The results from LION-A are a minimum free time of 25 minutes at an unscheduled time segment starting at time 347 and a maximum free time of 43 minutes at an unscheduled time segment starting at time 35. Now these values are compared with the previous incumbent values of minimum and maximum free time. Since 43 > 27, the new incumbent value of maximum free time is 43 minutes at RTS LION-A with an unscheduled time segment starting at time 35. The incumbent minimum free time value is not updated since the minimum value of 25 at RTS LION-A is greater than the incumbent minimum free time value of 23. Since neither of the first feasible insertion rules' stopping criteria is met, the current schedule at RTS COOK-B is examined for an insertion time. The results for this RTS found a minimum free time of 18 minutes at an unscheduled time segment starting at 429 and a maximum free time of 41 minutes at an unscheduled time segment starting at 112. Again these values are compared with the incumbent minimum and maximum free time values from the previous RTSs. A new incumbent minimum free time value is found since $18 < 23$. The incumbent maximum free time value stays the same since $43 > 41$. Finally, RTS POGO-A is examined for possible insertion times. The results for this RTS found a minimum free time of 21 minutes and a maximum free time of 39 minutes . The incumbent minimum and maximum free time values do not change since the minimum (maximum) value at RTS POGO-A is greater (less) than the incumbent minimum (maximum) free time value. Since all visible RTSs have been examined and neither of the stopping criteria have been met, the incumbent minimum and maximum free time values must be compared. Since the incumbent maximum free time is greater than thirty, it now becomes 13 $(43 - 30)$. Finally, since $13 < 18$, the best insertion time for support 255 is 35 at RTS LION-A (Support 255 has a start time of 50 $(35 + 15)$).

Schedule Improvement Algorithms

Overview. The schedule improvement algorithms attempt to reschedule a support or supports so that a previously unscheduled support can be added to the schedule. The basic methodology for the improvement algorithms is an interchange procedure. A two satellite interchange procedure is utilized to improve the number of low altitude satellites scheduled. A three interchange procedure is utilized to improve the number of medium/high altitude satellites scheduled. The general interchange methodology will first be explained followed by how this methodology is applied in the two and three satellite interchange procedures.

General Interchange Methodology. The general interchange methodology has three primary steps. These steps are performed in both the two satellite and three satellite interchange procedure. However, the mechanics of the steps will differ between the two and three satellite interchange procedures. The general steps of the procedure are presented below.

Step 1 Identify the RTSs the unscheduled support is visible to.

Step 2 Determine the set of scheduled supports that could be rescheduled.

Step 3 Determine if a scheduled support can be successfully rescheduled (called a successful interchange). A successful interchange occurs only if a free time block greater than or equal to that scheduled support's total support length can be found. If the scheduled support can be rescheduled, it is, and the unscheduled support is inserted into the now unscheduled time segment and RTS.

General Comments. Three comments can be made on how the general interchange methodology is applied to the two and three satellite interchange procedures. First, in the three satellite interchange procedure, two satellites will need to be rescheduled in order for a successful interchange to occur. Second, the application of Step 2 will differ depending on whether the unscheduled support is a low altitude support or medium/high altitude support. Medium/high altitude supports generally have a greater number of candidate scheduled supports within a RTS that can be rescheduled due to the flexibility in the start times of these supports. A rescheduling measure must be developed to determine which scheduled support is the best candidate support to be rescheduled. On the other hand, low altitude supports have no flexibility in their start times at a RTS, so a rescheduling measure is not required. Finally, for both the low and medium/high altitude unscheduled supports, the list of unscheduled supports is examined only once for possible addition to the schedule using an interchange procedure. If an unscheduled support cannot be added to the schedule, the support is not considered again for addition to the schedule. Therefore, both interchange procedures terminate after examining the last unscheduled support in the list.

Two Satellite Interchange Procedure.

The two satellite interchange procedure attempts to increase the the number of low altitude satellite supports scheduled after the supports are initially scheduled using the MIP linking procedure. The two satellite interchange procedure is performed after the MIP linking procedure and before the medium/high altitude insertion procedure. The two satellite interchange procedure attempts to schedule a previously unscheduled low altitude support by moving a scheduled support to an unscheduled period of time at another RTS. The steps of the two satellite interchange procedure are presented below. The following definitions are required to fully understand the steps of the two satellite interchange procedure. Conflicting supports are the supports scheduled at each visible RTS in the unscheduled support's visibility window. The flexibility of the conflicting support is defined as the number of RTSs each scheduled support is visible to. Each RTS the conflicting scheduled support is visible to, except the RTS the support is currently scheduled at, is called an alternate RTS.

Step 1 Identify all RTSs visible to the unscheduled support.

Step 2 Determine the scheduled supports that could be rescheduled. This step has two sub-steps: 1) determine the conflicting scheduled support at each of the visible RTSs and the flexibility of the conflicting support and 2) rank order conflicting supports by increasing flexibility. If an unscheduled support conflicts with two scheduled supports at a RTS, an attempt is made to reschedule both of the scheduled supports.

Step 2a Calculate the flexibility of each conflicting support

Step 2b Rank order conflicting supports by decreasing flexibility with the most flexible support ranked first and the least flexible support ranked last. Therefore, the most flexible scheduled support is the first candidate for the two satellite interchange.

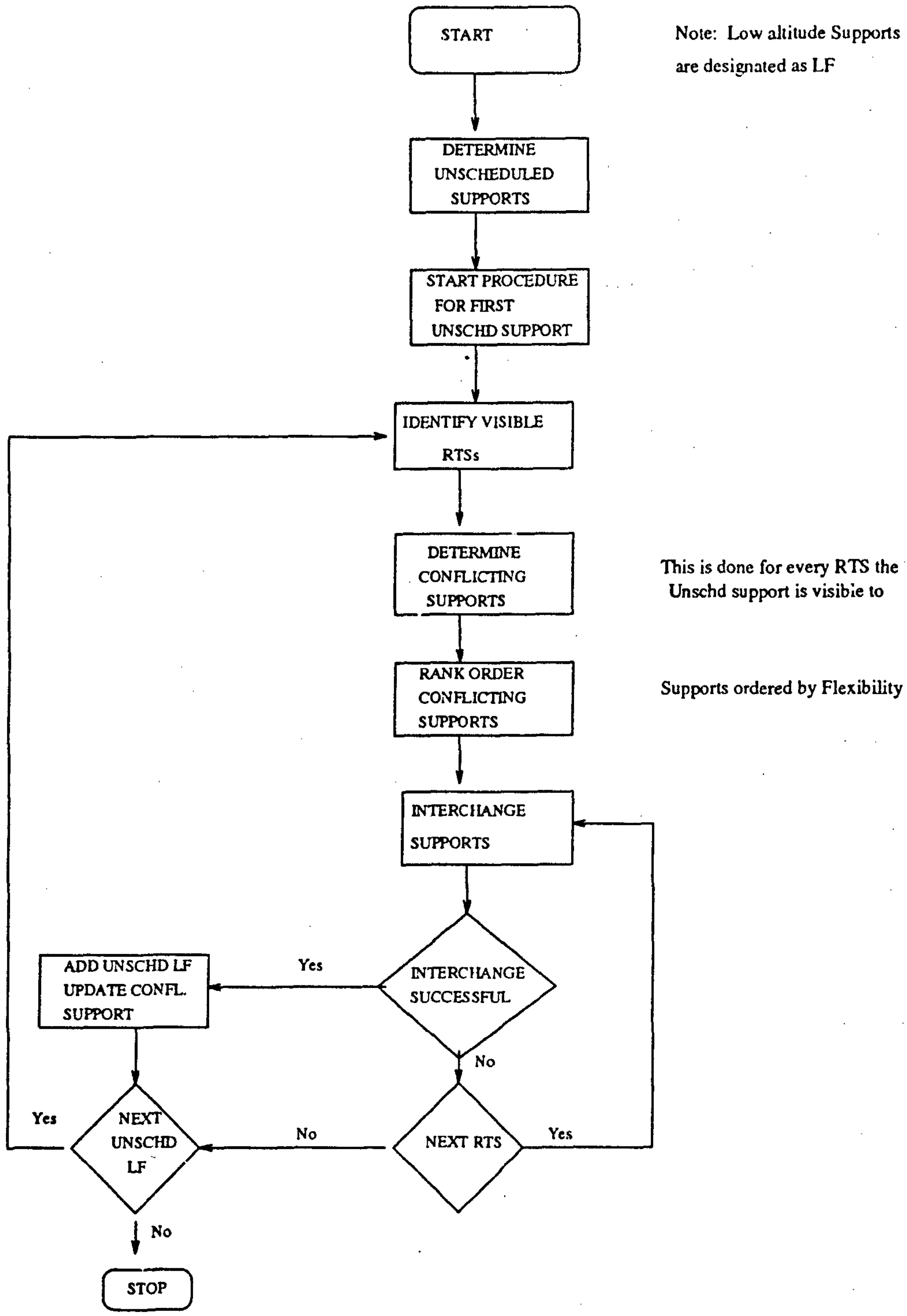

Figure 4.5. Flow of SRS Two Satellite Interchange Procedure

Step 3 The last step is to determine if any conflicting support on this list can be successfully rescheduled . For each alternate RTS, a check is made to determine if the conflicting support could be rescheduled at that RTS. A successful interchange occurs when no overlapping supports would occur at the alternate RTS when a conflicting support is rescheduled to that RTS. If a successful interchange occurs, the unscheduled support is scheduled in the vacated position, the conflicting support is rescheduled and the procedure does not consider any more conflicting supports. If a conflicting support cannot be rescheduled, the procedure then examines the next conflicting support in the list to determine if it can be successfully rescheduled. If all the conflicting supports have been checked and a successful interchange has not occurred, the low altitude satellite support remains unscheduled. After a successful interchange occurs or the support remains unscheduled, the procedure proceeds to the next unscheduled support in the list.

Two Satellite Interchange Example. The concepts of the two satellite interchange procedure will be illustrated with the following example. The data for the example is shown in Table B.1 In the example, it is assumed that support 8 is an unscheduled support, support 171 is scheduled at RTS POGO-A and support 95 is scheduled at RTS INDI-A. The two satellite interchange procedure will attempt to schedule support 8 by rescheduling either support 171 or 95 to an alternate RTS. Support 171 is visible to two alternative RTSs while support 95 is visible to one alternative RTS. The two interchange methodology will now be applied step by step for this support. Since support 8 is only visible to RTSs POGO-A and INDI-A, the above information completes the first two steps of the procedure; **Step 1** is determining the RTSs where the unscheduled support is visible and **Step 2a** is determining the conflicting scheduled supports at these RTSs and the flexibility of the conflicting supports. The next step, **Step 2b**, is to rank order the conflicting scheduled supports by flexibility. Support 171 is the most flexible since it is visible to three alternate RTSs compared to one alternate RTS for support 95. The final step, **Step 3,** is

Table 4.2. Two Satellite Interchange Example

Block 1 Data							
Sup No.	RTS	BV	EV	Req Len	TAT	Ident	Rev No
76	PIKE-A	642	647	5	20	2532	8073.1
76	POGO-C	643	658	15	15	2532	8073.2
289	INDI-A	643	644	1	20	9845	9291.0
24	LION-B	645	661	16	15	1056	9756.3
188	GUAM-A	645	653	8	20	6790	2683.0
272	BOSS-B	648	663	15	20	9757	773.3
272	PIKE-A	649	664	15	20	9757	773.3
55	BOSS-A	651	668	17	20	1748	9480.1
171	POGO-A	653	668	15	15	6553	3864.2

Block 2 Data							
Sup No.	RTS	BV	EV	Req Len	TAT	Ident	Rev No
8	POGO-A	654	671	17	20	286	2860.5
39	LION-A	654	667	13	20	1132	6856.1
8	INDI-A	657	661	4	15	4774	2583.5
55	POGO-B	660	676	16	20	1748	9480.2
289	POGO-C	663	679	16	15	9845	9291.2
95	INDI-A	667	679	12	20	3187	5786.9
171	BOSS-A	667	675	8	15	6553	3864.3
171	PIKE-A	667	682	15	20	6553	3864.3
188	POGO-C	667	683	16	15	6790	2683.2
95	GUAM-B	669	681	12	20	5821	4736.1
171	COOK-A	669	683	14	15	6553	3864.4

to determine if a scheduled support can be successfully rescheduled. Therefore, the procedure will first attempt to reschedule support 171 at one of its alternate RTSs. Each of the alternate RTSs (PIKE-A, COOK-A and BOSS-A) is checked to see if support 171 can be rescheduled at that RTS. If support 171 can be rescheduled, then the procedure stops and support 8 is now scheduled at POGO-A and support 171 is rescheduled at an alternate RTS. If support 171 cannot be rescheduled, then support 95 is examined. The interchange process is repeated by checking the alternate RTS (GUAM-B) for support 95. If support 95 can be rescheduled, then support 8 will be scheduled at RTS INDI-A. Otherwise, support 8 remains unscheduled.

Three Satellite Interchange Procedure

The three satellite interchange procedure attempts to increase the number of medium/high altitude satellite supports scheduled after the supports were initially scheduled using an insertion procedure The three satellite interchange procedure attempts to schedule a previously unscheduled medium/high altitude support by moving scheduled supports to alternate RTSs and time segments. Conceptually, the three satellite interchange procedure is one level higher than the two satellite interchange procedure. The two satellite interchange procedure cycles through the three steps of the general interchange methodology only once to determine if a successful interchange can occur. The three satellite interchange procedure cycles through the three steps of the general interchange methodology twice to determine if a successful interchange can occur. Figure 4.6 shows the flow of the three interchange procedure. Two definitions are required to fully understand the steps of the procedure. A **candidate support** is defined as a scheduled support that can be rescheduled and satisfies three criteria which are explained in Step 2a below. A **best candidate support** is the candidate support with the most potential to be rescheduled successfully.

Flow of Three Satellite Interchange Procedure As mentioned above, the three satellite interchange procedure cycles through the general steps of the interchange methodology two times. In fact, the three satelli' interchange is conceptually the same as two applications of the two satellite procedure. The first application determines the first best candidate support and checks to see if a successful interchange will occur. If a successful interchange does not occur, the two satellite interchange procedure is applied a second time to try to reschedule the best candidate support. If a successful interchange occurs in the second application, the schedule is updated to reflect the new start times of the previously unscheduled support and all rescheduled supports. However, the application of the three general steps differs between the two and three satellite interchange procedures. The three satellite interchange procedure applies to medium/high altitude satellites, while the two interchange procedure applies to low altitude satellites. In the three satellite interchange procedure, a medium/high altitude satellite support has greater scheduling flexibility due to longer visibility windows and a rescheduling flexibility measure is required to determine the best candidate support. In the two satellite interchange procedure, there is very little scheduling flexibility and the set of candidate supports is the list of conflicting supports. The steps of the three satellite interchange procedure are presented below.

For a single unscheduled support, labeled u, the following steps are executed.

Step 1 All the RTSs visible to u must be identified.

Step 2 Determine the best support that could be rescheduled. This process has two sub-steps: 1) determine candidate supports, and 2) determine the best candidate support.

Step 2a List each scheduled support that meets the following criteria and is currently scheduled at a time when support u could be scheduled. This list is called the **candidate support list** and every support on this list, labeled c, must meet three criteria: 1) the time block (free time around the candidate support plus

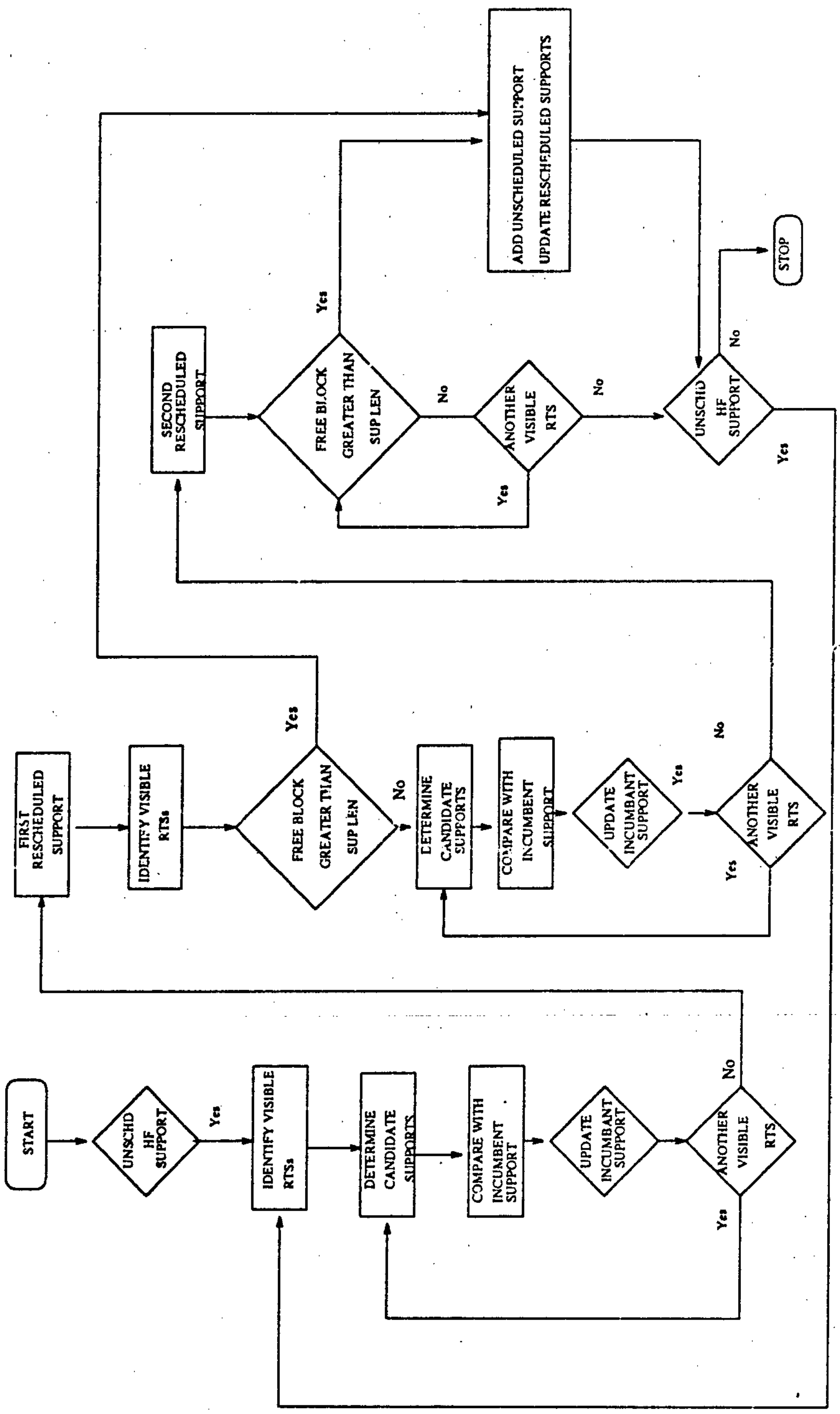

Figure 4.6. Flow of SRS Three Satellite Interchange Procedure

the candidate support's total support length) for each candidate support examined must be larger than support u's total support requirement. 2) support u's visibility requirements can be met during this time block and 3) the scheduled support is not a low altitude support or a major RTS downtime. These two types of supports are considered fixed and this procedure will not attempt to reschedule these.

Step 2b For each support c in the candidate support list, calculate a rescheduling flexibility measure. The flexibility measure is the product of the scheduling difficulty value times an average free time value for support c. The scheduling difficulty value is the inverse of the ratio used to rank order supports in the insertion procedure and is the ratio of the visibility window length to the total support length. The average free time value is calculated in a two step process. First, determine the average of the three largest free time values at each RTS support c is visible to. Second, sum these averages across the RTSs visible to support c. Mathematically, the rescheduling flexibility measure is computed in the following manner:

$$\frac{(\sum_{j \in RTS_c} FlexAvg_j)}{|RTS_c|} * \frac{VisAvg}{SupLen}$$

where:

$c=$ the candidate support

$j \in RTS_c$

$FlexAvg_j =$ the average of the three largest free time blocks at each visible RTS in support c's visibility window.

$SupLen =$ The total length of support c.

$VisAvg =$ average request window for support c and is defined as: $\frac{\sum_{j \in RTS_c} EV_{cj} - BV_{cj}}{|RTS_c|}$

After the candidate support's rescheduling flexibility measure is calculated, it must be compared to the best measure to date associated with the current incumbent support. The greater the flexibility measure, the more potential the candidate support has of being rescheduled. Therefore, if the new flexibility measure is greater than the current best value, the current candidate support becomes the incumbent candidate support and the current flexibility value is the new incumbent flexibility value. When all candidate supports have been examined, the incumbent candidate support becomes the best candidate support.

Step 3 Determine if a successful interchange can occur. In this step, a check is made to determine if the largest free time block found at any RTS visible to the best candidate support is greater than the best candidate support's total support length. If this condition is true, the best candidate support is rescheduled and the unscheduled support is scheduled. If a successful interchange does not occur and the second cycle through the three steps has not been completed, proceed to step 4. If the second cycle of the three steps has been completed, then the unscheduled support cannot be added to the schedule and the procedure proceeds to the next unscheduled support in the list.

Step 4 The first application of a two satellite interchange procedure was attempted using the steps described above and a successful interchange did not occur. A second application of the three steps is now attempted. The best candidate support for the first cycle now becomes the "unscheduled support" and the procedure proceeds through the three steps described above to determine the best candidate support for the second cycle and if a successful interchange can occur.

Three Satellite Interchange Example. The following example will illustrate the techniques utilized in the three satellite interchange procedure. Table 4.3 displays the support data for the example problem.

For illustration purposes, the Flexibility Average (Flex Avg) column is the mean of the average free time for all RTSs the candidate support is visible to ($FlexAvg = (\frac{\sum_{j \in RTS_c} FlexAvg_j}{|RTS_c|})$). The total support length (Tot Sup Len) column is the average of the support requirement plus the turnaround time. The time block

Table 4.3. Three Satellite Interchange Example

Unscheduled Support Information		
Sup No	# RTS visible to	Tot Sup Len
300	7	65

Candidate Supports for 1st Rescheduled Support						
Cand Sup	RTS	Tme Block	Flex Avg	Tot Sup Len	Vis Wind	Time Loc
289	INDI-A	75	85	60	400	602
324	LION-B	77	74	50	500	407

Free Time Blocks for 1st Rescheduled Support						
Cand Sup	RTS	Tme Block	Flex Avg	Tot Sup Len	Vis Wind	Time Loc
N/A	POGO-A	43				678
N/A	COOK-B	47				800
N/A	BOSS-A	46				805

Candidate Supports for 2nd Rescheduled Support						
Cand Sup	RTS	Tme Block	Flex Avg	Tot Sup Len	Vis Wind	Time Loc
279	LION-B	68	92	55	600	518
319	GUAM-A	66	65	60	320	744

Free Time Blocks for 2nd Rescheduled Support						
Cand Sup	RTS	Tme Block	Flex Avg	Tot Sup Len	Vis Wind	Time Loc
N/A	HULA-A	43				678
N/A	PIKE-A	56				743
N/A	POGO-B	47				800
N/A	COOK-A	46				805

column (Tme Block) is the sum of the Tot Sup Len column plus the free time adjacent to the candidate support. The visibility window (Vis Wind) column equals $EV_{ij} - BV_{ij}$. The time segment location column (Time Loc) shows the time in the day where the Tme Block column is located at. Additionally, two facts can be stated after examining the data: 1) the candidate support or time block is within the visibility window of the unscheduled or rescheduled support, 2) the RTS of the candidate support in the table is visible to the unscheduled or rescheduled support. The supports listed under the candidate support columns are the only ones that meet the three criteria for candidate supports. For example, in the Cand Sup column for first rescheduled support data, all seven RTSs the unscheduled support is visible to are examined and the two candidate supports listed are the only ones that meet the three criteria.

Determining 1st Rescheduled Support. Support 289 is the first candidate support examined to determine if it will be the first rescheduled support. The steps of the algorithm are applied to this support.

Step 2a: Support 289 meets all the criteria for a feasible candidate support.

Step 2b: Rescheduling flexibility measure calculation. The flexibility measure for support 289 is $85 * (400/60) = 566.6$.

Step 2b: Comparison. Since support 289 is the first candidate support, its flexibility measure is the best.

Support 324 will be examined next.

Step 2a: Support 324 meets all the criteria for a candidate support.

Step 2b: Rescheduling flexibility measure calculation. The flexibility measure for support 324 is $74 * (500/50) = 740$.

Step 2b: Comparison. Since 740 $>$ 566.6, support 324 becomes the incumbent candidate support. Since no other candidate supports were identified, support 324 is the first rescheduled support.

Step 3 Determine if successful interchange can be accomplished. Since none of the free time blocks listed for the first rescheduled support are larger than support 324's total support length, a successful interchange cannot occur.

Step 4 The second application of a two satellite interchange process will proceed with support 324 as the "unscheduled support".

Determining 2nd Rescheduled Support. Since support 324 was identified as the first rescheduled support, the second rescheduled support must now be determined. Again, each of the steps of the procedure will be applied to determine the second rescheduled support.

Step 1 At this step, all visible RTSs are identified. This step has already been accomplished when generating Table 4.3. Additionally, each RTS has been examined for a candidate support.

Step 2 is now applied to each candidate support listed in the candidate supports for 2nd rescheduled support section of Table 4.3 . Support 279 will be examined first.

Step 2a: Support 279 meets all the criteria for a feasible candidate support.

Step 2b: Rescheduling flexibility measure calculation. The flexibility measure for support 279 is $92 * (600/55) = 1003$.

Step 2c: Comparison. Since support 279 is the first candidate support, its flexibility measure is the incumbent flexibility measure.

Support 319 will be examined.

Step 2a: Support 319 meets all the criteria for a feasible candidate support.

Step 2b: Rescheduling flexibility measure calculation. The flexibility measure for support 319 is $65 * (320/60) = 346.6$.

Step 2b: Comparison. Since 1003 > 346.6, support 279 remains the incumbent candidate support. Since no other candidate supports were identified, support 279 is the second rescheduled support.

Step 3: Determine if a successful interchange can be accomplished. A free time block greater than 55 must be found for a successful interchange to occur. At RTS PIKE-A, a free time block of 56 was found, which indicates a successful interchange can occur. Now the unscheduled and two rescheduled supports are updated to reflect their new RTS and start times. Therefore, support 300 is added to RTS LION-B with a start time of 422 (407 + 15 minute turnaround time), support 324 is updated to GUAM-A with a start time of 759, and support 279 is updated to PIKE-A with a start time of 758.

SRS Heuristic Summary

In this chapter, the SRS heuristic approach was explained. The approach attempts to duplicate the scheduler logic in scheduling the most restrictive supports (low altitude satellite supports) first and then schedule the medium and high altitude supports around the low altitude supports. Additionally, the most difficult medium and high altitude supports are scheduled first and the supports with the shortest request length and longest visibility window are scheduled last.

The heuristic follows this logic with a two phase procedure that schedules as many low altitude supports as possible first and then schedules as many of the medium/high altitude supports as possible. In each phase, the schedule generation algorithm initially schedules as many satellite supports as possible. Then the schedule improvement procedure attempts to adjust the schedule so that unscheduled satellite supports can be added to the schedule. For low altitude satellites, a linked mixed integer programming approach is used to initially generate a schedule. A two satellite interchange procedure then attempts to add a previously unscheduled low altitude support to the schedule by rescheduling a scheduled low altitude support. For the medium and high altitude supports, an insertion procedure adds these supports to the schedule. After the insertion procedure is completed, a three satellite interchange procedure is performed to see if an unscheduled medium/high altitude support can be added to the schedule by rescheduling two previously scheduled medium/high altitude supports.

V. SRS Results

Overview

This chapter will examine the results of our testing of the SRS heuristic algorithm using real AFSCN data. Six data sets, containing the required satellite support information for the week of 12 Oct to 17 Oct 1992, were provided by Ken Chambers, technical advisor, 21 SOPS. Each data set was an ASCII file of the ASTRO general list database and represented a day of satellite support requirements. Both the satellite support requests and RTS downtime requirements were included when generating a 24 hour schedule. Four key areas will be discussed in this chapter:

- summary of results for each day
- comparison of results with those of the IBM study and with those of present day range schedulers
- analysis of unscheduled supports
- algorithm limitations

The summary of results will include statistics on the number of low and medium/high altitude satellite supports requested and those actually scheduled. This statistic provides an indication of how successful the algorithm was in scheduling supports. The comparison with the IBM study and current schedulers will examine the percent of supports scheduled by these two approaches versus the percent scheduled by our heuristic. The section on unscheduled supports will categorize unscheduled supports and provide an example of each category. The limitations section will discuss limitations of the algorithm and the types of satellite supports not included when generating a schedule.

Results

Overview. Overall, the results from the algorithm were very encouraging. For each data set, at least 90% of all satellite support requests were scheduled. Table 5.1 shows the percent of supports scheduled for each data set. Appendices C through G contain the generated schedule for each day and a listing of the unscheduled supports.

Table 5.1 Summary of Results, Percent of Supports Schedule

	Total # of Supports			Low Altitude Sup			High Altitude Sup		
Day	# Req	# Schd	% Schd	# Req	# Schd	% Schd	# Req	# Schd	% Schd
Day 1	352	324	92.0	153	146	95.4	199	176	88.4
Day 2	357	330	92.5	152	146	95.3	205	184	89.7
Day 3	371	341	92.0	156	150	96.1	215	191	89.0
Day 4	367	341	93.0	151	145	96.0	216	196	90.5
Day 5	361	329	91.1	154	149	96.7	207	180	87.0
Day 6	356	327	91.8	151	146	96.7	205	181	88.2

Comparison of Results with Current System and IBM Study

The results generated in this research effort will be compared with the current manual system and the IBM study. The results used in the comparison will be the percentage of supports scheduled and RTS utilization.

Percentage of Supports Scheduled. According to Francis Wong, Supervisor of Planning, 21 SOPS, who has 25 years of scheduling experience, the current schedulers successfully schedule between 95% and 98% of all supports requested in the initial 24 hour schedule(Wong, 28 January 1993). Therefore, between 2% and 5% of the supports are in conflict. The IBM study reported results for three days with 98% of the supports scheduled in the first day, 99% scheduled the second day and 92% of the supports scheduled in the third day. Table 5.2 shows the results of the IBM study.

Table 5.2. IBM Summary of Results

Day	# Req	# Schd	% Schd	# RTS antennas
Day 1	291	286	98	12
Day 2	292	289	99	12
Day 3	339	313	92	12

(Arbabi,1984:62) (Arbabi,1985:277)

The results generated by this research effort are slightly lower than the results of both the current schedulers and the IBM results. A possible reason for this is the hard time window and turnaround time assumption stated in Chapter I. The scheduling input data for this research is fixed, while in certain cases, the current range schedulers and IBM systems allowed slight modifications of the scheduling input data. Since there is no standard set of test data, a direct comparison between the three methods is not possible at this time.

Analysis of Unscheduled Supports

As shown in the summary of results section, approximately 8% of the supports each day were not scheduled. There are two reasons why a support is not scheduled:

- the support conflicts with scheduled supports and/or RTS downtime during its visibility window.
- a support whose beginning of visibility plus its request length is greater than the end of the day.

Support Conflicts with another Support/RTS Downtime In some cases, a support cannot be scheduled since all possible scheduling locations for this support overlap with previously scheduled supports and/or RTS downtime. In our testing, all unscheduled low altitude satellite supports fell into this category as well as some unscheduled medium/high altitude satellite supports. An example of an unscheduled medium/high altitude satellite in this category is support number 342 (IRON 4845), which was requested in day 5. Support 342 had a turnaround time of 15 minutes, a support length of thirty minutes, and was visible to INDI-A, REEF-A, LION-A and LION-B between 30 and 120. Support 342 could not be inserted into the schedule at any of the above RTSs for the following reasons. At RTSs INDI-A and LION-A, the support conflicted with RTS downtime and/or a scheduled low altitude support. At LION-B and REEF-A, the support met conflicted with scheduled medium/high supports and a scheduled low altitude support. The scheduled high altitude satellites at RTSs LION-B and REEF-A met at least one of the following two criteria: 1) a shorter support requirement length (the support could be inserted into a smaller time block) or 2) a higher support length to time window ratio (the support was inserted into the schedule before support 324) .

Beginning of visibility at the end of a day Some medium and high altitude supports have their beginning of visibility window start at the end of one day while the end of their visibility window falls in the next day. In some cases, the support has a long support requirement length such that the beginning of visibility plus support length is greater than the end of the current day. In this case, the SRS algorithm does not schedule the support since the algorithm treats each day independently and does not attempt to schedule supports across days. This problem will be discussed more in depth in the algorithm limitations section. An example of a support that could not be scheduled and falls into this category is support 502. This support had a support length of one hour and a beginning of visibility time of 1402. Support 502 could not be scheduled since its support length plus visibility time (1402 + 60 > 1440) overlapped into the next day. (Note: 2400 in hhmm fcrmat equals 1440 when the day is depicted in minutes)

Algorithm Limitations

There are two primary limitations that prevent this algorithm from attempting to schedule every type of support. First, special case supports were not attempted.

Examples of special case supports would be a hot-handoff support request, an interval request, and a simultaneous support request. A hot-handoff request is where an extremely long support must be serviced by more than one RTS antenna and the support must be "handed off" from one antenna to the next. An interval support request is where a satellite must have supports every "X" hours. A simultaneous support is where a single satellite is supported on two RTS antennas simultaneously. Special case supports do not occur on a regular basis and these supports would most likely be scheduled before any other medium/high altitude satellite supports. Finally, the algorithm looks at only one day at a time when scheduling supports. Therefore, the algorithm does not attempt to schedule a support across days, i.e. when the support begins at the end of one day and is completed in the beginning of the next day. Suggestions on how to correct these limitations are presented in Chapter VI.

VI. Conclusions and Recommendations

Conclusions

Satellite range scheduling (SRS) is a complex problem that involves scheduling satellite supports in which a satellite and a specific remote tracking station are assigned a time window during which they communicate with each other. As the number and complexity of satellite supports continue to increase, more pressure is placed on the current manual system to efficiently schedule all possible supports. Therefore, finding ways to automate all or sections of the current satellite range scheduling process will decrease the time required to generate a schedule and increase the range scheduler's capability. The objective of this research effort was to develop a methodology to automate the initial 24 hour schedule. The goal of the scheduling algorithm was to maximize the number of conflict free satellite supports scheduled. The methodology developed was a two pronged approach of schedule generation algorithms and schedule improvement algorithms. The schedule generation algorithms were a MIP linking procedure for low altitude satellites and an insertion procedure for medium/high altitude satellites. The schedule improvement algorithms were a two satellite interchange algorithm for low altitude satellites and a three satellite interchange procedure for medium/high altitude satellites. The heuristic uses a two phased approach to generate a schedule. For each phase, a schedule is first generated using the schedule generation algorithm and then the schedule improvement algorithm attempts to schedule an unscheduled support by rescheduling a scheduled support. In the first phase, low altitude satellites are scheduled using the MIP-linking procedure and the two satellite interchange algorithms. In the second phase, medium/high altitude satellites are scheduled using the insertion procedure and the three satellite interchange algorithm.

A schedule was generated for six representative data sets with promising results. For each data set, at least 91% of all satellite supports were scheduled. These results are slightly lower than the results of the current range schedulers and the results of a previous IBM study. A possible reason for this difference is the assumption in this research effort that all scheduling inputs are fixed. Based on the results reported, the methodology presented in this research effort seems to be a valid approach for automating the initial 24 hour schedule. The primary limitation of the algorithm is its inability to schedule every type of satellite support. Supports that need to be scheduled across days and special case support requests are not scheduled in the current algorithm. These support requests would have to be manually scheduled if the algorithm was implemented in its current state. Suggestions are made in the following recommendations section for possible enhancements of the algorithm so that all support types can be scheduled.

Recommendations

This section concentrates on four areas: 1) further testing of the current algorithm, 2) additional research to enhance the current scheduling algorithm, 3) suggestions on how to eliminate some of the limitations of the algorithm and 4) suggestions on how to standardize testing for future SRS automation efforts.

The first recommendation is that further testing of the current algorithm should take place to develop a better understanding of the algorithm's capabilities and limitations. The algorithm should be tested on additional data sets to 1) verify the algorithm's stated capabilities and limitations and 2) determine if any additional capabilities or limitations of the algorithm exist.

The second recommendation is that further investigation should take place to determine what enhancements of the current algorithm could be accomplished. Two possible enhancements would be to expand the search for candidate supports in the three satellite interchange procedure and to develop algorithms to schedule special case supports. The first possible enhancement of the current algorithm would be to expand the search for candidate supports in the high altitude three satellite interchange procedure. The current procedure examines only one candidate support, based on the best scheduling flexibility measure. Expanding the number of candidate supports would increase the execution time of the algorithm but also increase the chance that an unscheduled support can be scheduled. The second possible enhancement would be to develop algorithms to handle the special case requests. Special cases, by their nature, do not occur on a regular basis and have a high priority. Therefore, these cases would be scheduled before any medium/high altitude supports and in some cases preempt low altitude satellite supports. An example of this would include a procedure that allows high altitude satellites with long support requirements to divide the support up so that two or more RTSs would perform a "hot handoff". In this procedure, if the entire support could not be scheduled at one RTS, the support would be divided up into segments and "hot handoffs" would occur between the RTSs the support was scheduled at. Currently, schedulers have to manually schedule special case supports. It is not clear whether an automated approach would significantly aid in the scheduling process for these supports.

The third recommendation would be to correct the limitation of not being able to schedule a support over more than one day. This could be accomplished by building a partial two day schedule. In this procedure, a partial schedule would be built for the second day that consisted of only known low altitude supports. Under this approach, the flow of the scheduling algorithm would be changed in the following manner. The MIP linking procedure and two satellite interchange procedure for a given day (labeled day 1) would be accomplished exactly as they are now. However, before the medium/high altitude supports are attempted to be inserted into the schedule (for day 1), known low altitude supports for the next day (day 2) would be scheduled using the MIP approach. The medium/high altitude supports would then be inserted into the schedule for both day 1 and day 2. The only medium/high altitude supports scheduled in day 2 would be the supports that were scheduled for both days. For example, in this approach, a support that had a two hour support requirement, but had a beginning of visibility time of 2300 could now have a start time of 2330 on day 1 with the support ending at 0130 on day 2 at a given RTS. Then when the 24 hour schedule for day 2 is developed, these scheduled supports must be accounted for so that overlapping supports are avoided.

The last recommendation concerns the automation of the satellite range scheduling process. According to Lt Greg Schultz, SRS project officer, a program that automates a portion or all of the SRS scheduling process has a tentative completion deadline of early 1996 (Schultz: 11 Feb 1993). To facilitate comparison between automation efforts, a set of test data problems should be developed. Therefore, the different automated scheduling programs would generate a schedule for each data set and the results of each program could be objectively compared.

Appendix A. SRS Modeling Software.

Overview

The SRS model was developed using three software packages/ programming languages. The computer code for the data input table program and linking procedure were written in FORTRAN. The MIP formulation was modeled with The General Algebraic Modeling Software (GAMS). The interchange procedures and insertion procedure were programmed in Borland Turbo Pascal 6.0. Both the FORTRAN programming code and the GAMS software were run on a VAX/VMS 6420 mainframe computer. GAMS has a PC version of its software that has the same capability as the mainframe version. The Turbo Pascal software was run on a UNISYS *PW*^{2}*Advantage* PC computer. The GAMS software was chosen to create the MIP model of the SRS problem for two reasons: 1) data input tables can be created off line and read into the model and 2) the constraints of the model are represented algebraically in GAMS. The Turbo Pascal 6.0 was chosen because of its code generation and debugging features, and its compatibility with the existing ASTRO system.

This appedix contains information in the following areas:

- FORTRAN MIP Linking Procedure Code
- GAMS programming code
- sample GAMS data input table
- Turbo Pascal code

FORTRAN & GAMS Code

FORTRAN Code. The FORTRAN code was developed to 1) generate input tables for GAMS, and 2) link MIP blocks together. The code used to perform these functions is shown below:

```
c*************************************************************
c   PROGRAM NAME: IPLINK
c    REV    DATE OF
C    NO.    CHANGE     DESIGNER     CCR   DESCRIPTION
C    ---    ------     --------     ---   -----------
C    000    12NOV92    T.GOOLEY            ORIGNIAL RELEASE
c
c
c   Program Description: Links results of 1st MIP run
c    to the next MIP run
c
c    Indices:  I- support No.:  J- RTS
c
c   Array Description:
c
c    X(I,J)    - holds the feasible combinations of support/RTS
c    ABV(I,J) - holds beginning of visibility information
c    AEV(I,J) - holds end of visibility information
c    R(i,j)    - holds support length requirments
c    W(i)      - holds weight or flexibilty value
c    VAL(I,J) - holds the MIP schedule from the last
c                 MIP block
c    DOWN(j)  - holds the lastest RTS resource usage
c                from last block
c    AGS(j)    -  holds the RTS names: POGO-A, etc.


      PROGRAM SORT
      INTEGER  I,J,K,REQ,NUM,INDEX,IREV
      INTEGER,INREV,SNUM,NUM5,SCHD(150)
      CHARACTER*78 line,LINE1
      CHARACTER*3 GGTS(18)
      CHARACTER*6 GTS,crev,AGS(18)
      CHARACTER*4 CBV1,CEV2,CREQ
      INTEGER TA,BV,EV,BST,EST,ID,IDENT,csum,MAX,sdt,FL
      INTEGER TO(150),D1,NWI(400),ST(150,18),VAL(150,18)
      REAL REV,W(150),V(150,18)
      INTEGER R(150,18), X(150,18),ABV(150,18),AEV(150,18)
      INTEGER CNT, CNT1,DOWN(18),FL1,RE(150,18)
      IREV=0.0
      OPEN(UNIT=10,FILE='SUP.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=9,FILE='SINPUT1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=19,FILE='SUP1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=12,FILE='NTABLE1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=22,FILE='avg.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=21,FILE='HOLD.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=11,FILE='HOLD1.DAT',STATUS='UNKNOWN')
      OPEN(UNIT=32,FILE='sch.DAT',STATUS='UNKNOWN')
```

```
OPEN(UNIT=31,FILE='strt.DAT',STATUS='UNKNOWN')
OPEN(UNIT=33,FILE='REQ.DAT',STATUS='UNKNOWN')
OPEN(UNIT=99,FILE='TEMP.DAT',STATUS='UNKNOWN')


I=0
IDENT=0
AGS(1)='POGO-A'
AGS(2)='POGO-B'
AGS(3)='POGO-C'
AGS(4)='POGO-D'
AGS(5)='HULA-A'
AGS(6)='HULA-B'
AGS(7)='COOK-A'
AGS(8)='COOK-B'
AGS(9)='INDI-A'
AGS(10)='INDI-B'
AGS(11)='BOSS-A'
AGS(12)='BOSS-B'
AGS(13)='LION-A'
AGS(14)='LION-B'
AGS(15)='GUAM-A'
AGS(16)='GUAM-B'
AGS(17)='PIKE-A'
AGS(18)='REEF-A'
GGTS(1)='P-A'
GGTS(2)='P-B'
GGTS(3)='P-C'
GGTS(4)='P-D'
GGTS(5)='H-A'
GGTS(6)='H-B'
GGTS(7)='C-A'
GGTS(8)='C-B'
GGTS(9)='I-A'
GGTS(10)='I-B'
GGTS(11)='B-A'
GGTS(12)='B-B'
GGTS(13)='L-A'
GGTS(14)='L-B'
GGTS(15)='G-A'
GGTS(16)='G-B'
GGTS(17)=' PI'
GGTS(18)='REF'
DO 11 J=1,150
DO 12 K=1,18
  NWI(J)=0
  X(J,K)=0
  ABV(J,K)=0
  AEV(J,K)=0
  RE(J,K)=0
```

```
       W(J)=0
       R(J,K)=0
       TO(J)=0
       DOWN(K)=0
       SCHD(J)=0
12    CONTINUE
11    CONTINUE

C
C  LOAD DATA FILES FROM LAST IP RUN
C

        MAX=0
 5      READ(10,FMT=99,ERR=111,END=111) SNUM,I,GTS
 99     FORMAT(I6,2X,I6,1X,A6)
        IF (I .GT. MAX) MAX=I
        GOTO 5

111    DO 77 J =1,5
        READ(32,FMT=123) LINE
        READ(33,FMT=123) LINE
        READ(31,FMT=123) LINE
123     FORMAT(A78)
 77    CONTINUE



      DO 21 J = 1, MAX
       READ(31,FMT=32) K,(ST(J,K),K=1,10)
       READ(33,FMT=32) K,(RE(J,K),K=1,10)
 21   CONTINUE


 32   FORMAT(I4,1X,10I5)

      DO 7 J =1,3
       READ(31,FMT=123) LINE
       READ(33,FMT=123) LINE
7     CONTINUE



      DO 41 J = 1, MAX
       READ(31,FMT=31) K,(ST(J,K),K=11,18)
       READ(33,FMT=31) K,(RE(J,K),K=11,18)
 41   CONTINUE
 31   FORMAT(I4,1X,8I5)



      DO 1 J = 1, MAX
```

```
      READ(32,FMT=2) K,(V(J,K),K=1,10)
 1    CONTINUE
 2    FORMAT(I4,1X,10F5.1)



      DO 666 J =1,3
       READ(32,FMT=123) LINE
666   CONTINUE


      DO 4 J = 1, MAX
       READ(32,FMT=3) K,(V(J,K),K=11,18)
 4    CONTINUE
 3    FORMAT(I4,1X,8F5.1)


      DO 46 J = 1,MAX
      DO 47 K = 1,18
       VAL(J,K)= NINT(V(J,K))
 47   CONTINUE
 46   CONTINUE

       DO 1110 J= 1,MAX
         WRITE(99,409) J,(VAL(J,K),K=1,11)
1110   CONTINUE

        DO 1120 J= 1,MAX
        WRITE(99,419) J.(VAL(J,K),K=12,18)
1120      CONTINUE

       DO 2110 J= 1,MAX
         WRITE(99,409) J,(ST(J,K),K=1,11)
2110   CONTINUE

        DO 2120 J= 1,MAX
        WRITE(99,419) J,(ST(J,K),K=12,18)
2120      CONTINUE


C
C    END OF DATA LOAD SEGMENT
C


C
C    DETERMINE THE LATEST TIME GTS J WAS USED IN LAST IP
C

      DO 36 K = 1,18
```

```
       DO 34 J= 1,MAX
          IF(VAL(J,K) .GT. .5 ) THEN
           SDT=0
           SDT=ST(J,K)+RE(J,K)+20
           IF (DOWN(K) .LT. SDT) DOWN(K) =SDT
          ENDIF
 34    CONTINUE
 36    CONTINUE
C
C        END GTS SECTION
C

C
C   DETERMINE SUPPORTS THAT WERE SCHEDULED IN LAST IP RUN


        REWIND(10)

  6     READ(10,FMT=99,ERR=112,END=112) SNUM,I,GTS

        NUM5=0
        DO 8 J= 1,18
         NUM5=NUM5+VAL(I,J)
 8      CONTINUE

        IF (NUM5 .GT. 0) SCHD(I)=SNUM
        GOTO 6

C
C      END OF SUPPORT SEGMENT
C
112         I=0

C
C  READ IN DATA FROM  FILE OF SUPPORTS THAT BV WAS LATE THAN
C  LAST TIME
C

10     READ(21,FMT=98,ERR=200,END=200) SNUM,GTS,bv,ev,req,
     * TA,IDENT,REV
 98    FORMAT(I4,1X,A6,I5,I5,I5,1X,I2,1X,I4,1X,F6.1)


       FL=0
       K=0
      DO 35 J = 1,18
        IF (GTS .EQ. AGS(J)) THEN
          K=J
        ENDIF
 35    CONTINUE
```

```
      IF (K .EQ. 0) GOTO 10
      IF (REQ .LT. 10) GOTO 10

C
C   CHECK SUPPORT AGAINST SCHEDULED SUPPORTS FROM LAST TIME
C    IF SUPPORT WAS SCHEDULED LAST TIME  SKIP IT
C
      DO  122 J = 1, MAX
         IF (SCHD(J) .EQ. SNUM) FL=1
 122     CONTINUE
      IF (FL .EQ. 1) GOTO 10

       IF (I .GE. 75) THEN
        WRITE(11,98) SNUM,GTS,bv,ev,req,TA,IDENT,REV
        GOTO 10
      ENDIF



C
C       END OF SUPPORT CHECK
C


       FLAG=0
       cnt=1
       cnt1=1
       DO 23 J = 1,150
         IF (NWI(J) .GT. 0) CNT1=CNT1+1

         IF (SNUM .EQ. NWI(J)) THEN
             CNT=J
             FLAG=1
         ENDIF
23       CONTINUE

         IF ( FLAG .EQ. 1) THEN
            I=CNT
         ELSE
             I=CNT1
             NWI(I)=SNUM
          ENDIF

        WRITE(19,99) SNUM,I,GTS

     K=0
     DO 25 J = 1,18
       IF (GTS .EQ. AGS(J)) THEN
         K=J
       ENDIF
 25    CONTINUE
       IF (K .EQ. 0) GOTO 10
```

```
          X(I,K)=1
          ABV(I,K)=BV
          AEV(I,K)=EV
          TO(I)=TA
          R(I,K)=REQ




C
C READ NEXT RECORD
C
        GOTO 10
C
C CLOSE FILES
C
C
C DETERMINE WEIGHT OF VARIABLE
 200  Csum=0
        DO 22 J= 1,I
          SUM=0
          DO 211 K=1,18
           SUM=SUM+X(J,K)
211       CONTINUE
         IF (SUM .GT. 0) THEN
           W(J)=1/SUM
         ENDIF
         csum=csum+sum
 22     CONTINUE
         write(22,97)  i,csum
 97      FORMAT(I5,2X, I8)


c
c  Determine overlap in GTS usage from last GAMS run
c
         DO 13 K=1 ,18
         DO 911 J=1, I
           IF (DOWN(K) .GT.  ABV(I,K) .AND. ABV(I,K) .GT. 0) THEN
                PRINT *, 'ABV = ',ABV(I,K)
                PRINT *, 'DOWN = ',DOWN(K)
                PRINT *, ' I = ',I,'   K = ',K
                X(I,K)=0
           ENDIF
 911      CONTINUE
 13      CONTINUE


C
C        WRITE TABLES TO FILE
C        CREATE GTS HEADING
```

```
c
      WRITE(12,102)
 102  FORMAT(5X,'SETS')
 103  FORMAT('  I   supports   /1*',I3,'/')
      WRITE(12,103) I
      WRITE (12,900)
 900  FORMAT('                   ')
      WRITE (12,105)
 105  FORMAT(7X,'J GTS')
      DO 19 J = 1,18
      IF (J .EQ. 1) THEN
        WRITE(12,106) GGTS(1)
 106    FORMAT(8X,'/',A3)
      ENDIF
      IF (J .EQ. 18) THEN
        WRITE(12,110) GGTS(18)
 110    FORMAT(10X,A3,'/;')
      ENDIF
      IF (J .GT. 1 .AND. J .LT. 18) THEN
          WRITE(12,107)  GGTS(J)
 107      FORMAT(10X,A3)
      ENDIF
  19  CONTINUE
      WRITE (12,900)
      WRITE(12,104)
 104  FORMAT(2X,'ALIAS(I,H);')
      WRITE(12,900)
      WRITE(12,991)
 991  FORMAT('SET OFFDIAG(I,H);')
      WRITE(12,992)
 992  FORMAT('OFFDIAG(I,H)=YES$(ORD(I) ne ORD(H));')
      WRITE(12,900)
      WRITE(12,997)
 997  FORMAT('SCALAR M large positive constant /5000/;')
      WRITE (12,900)
      WRITE (12,201)
c
c         Write TAT and Weight for each Support
c
201   FORMAT('PARAMETERS')
      WRITE (12,900)
      WRITE(12,302)
 302  FORMAT('  TO(I) turnaround time of a support')
      DO 59 J = 1 ,I
        IF (J .EQ. 1) THEN
          WRITE (12,303) J, TO(J)
303       FORMAT(4X,'/',I4,2X,I3)
        ENDIF
        IF (J .GT. 1 .AND. J .LT. I) THEN
          WRITE (12,304) J,TO(J)
```

```
304       FORMAT(5X,I4,2X,I3)
        ENDIF
        IF (J .EQ. I) THEN
          WRITE (12,305) J,TO(J)
305       FORMAT(5X,I4,2X,I3,' /;')
        ENDIF
 59   CONTINUE
      WRITE(12,900)
      WRITE(12,900)
      WRITE(12,402)
c
c      Write Support Requirement Table for each support
c
 402  FORMAT('TABLE R(I,J) request length supports of i and j')
      WRITE(12,900)
      DO 48 K=1,1
         WRITE(12,411) (GGTS(J),J=1,11)
 411     FORMAT(5X,11A6)
  48  CONTINUE
      DO 45 J = 1, I
        WRITE(12,409) J,(R(J,K),K=1,11)
  45  CONTINUE
 409  FORMAT(I4,1X,11I6)
      DO 466 K=1,1
         WRITE(12,421) (GGTS(J),J=12,18)
 421     FORMAT(3X,'+',1X,7A6)
 466  CONTINUE
      DO 467 J = 1, I
        WRITE(12,419) J,(R(J,K),K=12,18)
 467  CONTINUE
 419  FORMAT(I4,1X,7I6)
      WRITE (12,401)
 401  FORMAT(2X,';')
      WRITE(12,900)
      WRITE(12,108)
c
c     Write Feasible Support /RTS combinations table
c
 108  FORMAT('TABLE N(I,J) feasible supports of i and j')
      WRITE(12,900)
      DO 28 K=1,1
         WRITE(12,411) (GGTS(J),J=1,11)
  28  CONTINUE
      DO 29 J = 1, I
        WRITE(12,409) J,(X(J,K),K=1,11)
  29  CONTINUE
      DO 38 K=1,1
         WRITE(12,421) (GGTS(J),J=12,18)
  38  CONTINUE
      DO 39 J = 1, I
        WRITE(12,419) J,(X(J,K),K=12,18)
```

```
 39  CONTINUE
     WRITE (12,901)
 901 FORMAT(2X,';')
     WRITE(12,900)
     WRITE(12,508)
c
c    Write Beginning of Visibility Table
c
 508 FORMAT('TABLE BV(I,J) begining of  visibility')
     WRITE(12,900)
     DO 52 K=1,1
        WRITE(12,411) (GGTS(J),J=1,11)
  52 CONTINUE
     DO 53 J = 1, I
       WRITE(12,409) J,(ABV(J,K),K=1,11)
  53 CONTINUE
     WRITE(12,900)
     DO 62 K=1,1
        WRITE(12,421) (GGTS(J),J=12,18)
  62 CONTINUE
     DO 63 J = 1, I
       WRITE(12,419) J,(ABV(J,K),K=12,18)
  63 CONTINUE
     WRITE (12,901)
     WRITE (12,900)
     WRITE(12,608)
c
c   Write End of Visibility table
c

 608 FORMAT('TABLE EV(I,J) ending of  visibility')
     WRITE(12,900)
     DO 78 K=1,1
        WRITE(12,411) (GGTS(J),J=1,11)
  78 CONTINUE
     DO 79 J = 1, I
       WRITE(12,409) J,(AEV(J,K),K=1,11)
  79 CONTINUE
     WRITE(12,900)
     DO 88 K=1,1
        WRITE(12,421) (GGTS(J),J=12,18)
  88 CONTINUE
     DO 89 J = 1, I
       WRITE(12,419) J,(AEV(J,K),K=12,18)
  89 CONTINUE
     WRITE (12,901)
     write(12,900)
     WRITE(12,994)
 994 FORMAT('SET D(I,H,J);')
     WRITE(12,995)
 995 FORMAT('D(I,H,J)=YES$N(H,J)$N(I,J)$OFFDIAG(H,I);')
```

```
      WRITE(12,996)
996   FORMAT('D(I,H,J)$((BV(I,J) GT EV(H,J) + TO(I)) OR
     *(EV(I,J) +TO(H) LT BV(H,J))) =NO;')
      WRITE(12,900)
300   PRINT *,' NUMBER OF SUPPORT IS',I
       CLOSE(10)
       CLOSE(9)
       CLOSE(19)
       CLOSE(33)
       CLOSE(32)
       CLOSE(31)
       CLOSE(22)
       CLOSE(21)
       CLOSE(12)
       CLOSE(41)
       CLOSE(42)
      END
```

GAMS Code. The following GAMS model was developed and run for each SRS MIP block

```
c*************************************************************************
c
c   PROGRAM NAME:  SRSL1.GMS
c
c   Purpose:  To generate the GAMS SRS MIP model for one block of data
c
c  The file ntable.dat was created off line and contains all the set
c  definiations, parameter values and input tables required for the
c  GAMS model to run


$include "ntable.dat"


sets jj(j) dynamic subset of i to hold columns for subtable
     cc(j) dynamic subset of i to hold unprinted columns
     s     subtables / 1*12 /;
scalar maxcol;



VARIABLES

ST(I,J)  start time for support i at GTS j
X(I,J)   support i and GTS j 1 if support occurs 0 otherwise
Y(H,I,J)  relax or enforce constraint for supports h i and GTS j
Z         total weighted number of supports scheduled
;

POSITIVE VARIABLE ST;
BINARY VARIABLES X,Y;

EQUATIONS

SCH  obj function - weighted number of supports scheduled
SUPONE(I)  schedule support only once
BEGSUP(I,J)  schedule support after its beginning visibility
ENDSUP(I,J)  schedule support before its end of visibility
NCCSUP1(J,I,H)  no concurrent supports on a GTS j ST_i lt ST_j
NCCSUP2(J,I,H) no concurrent supports ST i GT ST h

;

SCH..  Z =E= SUM((I,J)$N(I,J), X(I,J));
```

```
SUPONE(I).. SUM(J$N(I,J), X(I,J)) =L= 1;

BEGSUP(I,J)$N(I,J)..  ST(I,J) =G= BV(I,J)*X(I,J);

ENDSUP(I,J)$N(I,J)..  ST(I,J) =L= (EV(I,J)-R(I,J))*X(I,J);


NCCSUP1(J,I,H)$D(I,H,J)..  ST(I,J) + R(I,J) + TO(H) =L=
                          ST(H,J) +M*Y(H,I,J) + M*(1-X(I,J))
                          + M*(1-X(H,J));
NCCSUP2(J,I,H)$D(I,H,J)..  ST(I,J) =G= ST(H,J) + R(H,J) + TO(I)
                         -M*(1-Y(H,I,J))-M*(1-X(I,J))-M*(1-X(H,J));

MODEL SCHEDULE  SRS Scheduling Solution /ALL/;

OPTION ITERLIM=5000000;

OPTION RESLIM=100000;

SOLVE SCEDULE USING MIP MAXIMIZING Z;

DISPLAY ST.L, X.L;


C*************************************************************
c
c  The following commands are used to write the results of a
c   GAMS MIP run into formateed ASCII files
c

file res /scb.dat/;

res.pw = 78;
put res '   table x(i,j) this is a table of scheduled supports'/;
jj(j) = no;
cc(j) = yes;
loop(s$card(cc),
   maxcol=floor(res.pw/7-1);
   loop(cc$maxcol,
      jj(cc) = yes;
      maxcol=maxcol-1);
   if((card(cc) ne card(j)), put // '+':6);
   if((not(card(cc) ne card(j))), put // ' ':6 );
   loop(jj, put jj.tl:>5); put /;
   loop(i,
      put / i.tl:5;
      loop(jj, put x.L(i,jj):5:0) );
   cc(jj) = no;
   jj(jj) = no );
put$card(cc) // '**** more than ' card(s):0:0 ' subtables'
              / '**** ' card(cc):0:0 ' columns not written';
```

```
abort$card(cc) 'not all columns were printed' , cc;

file res1 /strt.dat/;

res1.pw = 78;

put res1 '   table st(i,j) this is a table of support start times'/;
jj(j) = no;
cc(j) = yes;
loop(s$card(cc),
   maxcol=floor(res1.pw/7-1);
   loop(cc$maxcol,
      jj(cc) = yes;
      maxcol=maxcol-1);
   if((card(cc) ne card(j)), put // '+':6);
   if((not(card(cc) ne card(j))), put // ' ':6 );
   loop(jj, put jj.tl:>5); put /;
   loop(i,
      put / i.tl:5;
      loop(jj, put st.L(i,jj):5:0) );
   cc(jj) = no;
   jj(jj) = no );
put$card(cc) // '**** more than ' card(s):0:0 ' subtables'
              / '**** ' card(cc):0:0 ' columns not written';
abort$card(cc) 'not all columns were printed' , cc;


file res2 /req.dat/;

res2.pw = 78;

put res2 '   table r(i,j) this is a table of support request times'/;
jj(j) = no;
cc(j) = yes;
loop(s$card(cc),
   maxcol=floor(res2.pw/7-1);
   loop(cc$maxcol,
      jj(cc) = yes;
      maxcol=maxcol-1);
   if((card(cc) ne card(j)), put // '+':6);
   if((not(card(cc) ne card(j))), put // ' ':6 );
   loop(jj, put jj.tl:>5); put /;
   loop(i,
      put / i.tl:5;
      loop(jj, put R(i,jj):5:0) );
   cc(jj) = no;
   jj(jj) = no );
put$card(cc) // '**** more than ' card(s):0:0 ' subtables'
              / '**** ' card(cc):0:0 ' columns not written';
abort$card(cc) 'not all columns were printed' , cc;
```

GAMS Input Table. The GAMS input tables are the output of the FORTRAN code displayed above and an input file into the GAMS MIP model. There were five primary input tables required for the MIP formulation: 1) the feasible combinations of supports and RTS (what RTS is visible to each support), 2) the turnaround time for each support, 3) the support requirement time, and 4) the beginning visibility of each support and 5) the ending visibility of each support. Each record in the database was read in and the appropriate information for each input table was extracted. The format of the input tables had to look exactly like a table hand typed into the GAMS software. A GAMS error occurred if the input tables differed from the specified GAMS format. A sample input table is shown below:

```
1
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
C O M P I L A T I O N


INCLUDE: GOR93M:[TGOOLEY]NTABLE.DAT;1
        SETS
  I   SUPPORTS   /1* 85/

          J GTS
          /P-A
            P-B
            P-C
            P-D
            H-A
            H-B
            C-A
            C-B
            I-A
            I-B
            B-A
            B-B
            L-A
            L-B
            G-A
            G-B
             PI
```

```
          REF/;

  ALIAS(I,H);

SET OFFDIAG(I,H);
OFFDIAG(I,H)=YES$(ORD(I) NE ORD(H));

SCALAR M LARGE POSITIVE CONSTANT /5000/;

PARAMETERS


  TO(I) TURNAROUND TIME OF A SUPPORT
    /   1   20
        2   20
        3   20
        4   20
        5   20
        6   20
        7   20
        8   20
        9   20
       10   20
       11   20
       12   20
       13   20
       14   20
       15   20
       16   20
       17   20
       18   20
       19   20
       20   20
       21   20
       22   20
       23   20
       24   20
       25   20
       26   20
       27   20
       28   20
       29   20
       30   20
       31   20
       32   20
       33   20
       34   20
       35   20
       36   20
       37   20
       38   20
```

162	39	20

163	40	20
164	41	20
165	42	20
166	43	20
167	44	20
168	45	20
169	46	20
170	47	20
171	48	20
172	49	20
173	50	20
174	51	20
175	52	20
176	53	20
177	54	20
178	55	20
179	56	20
180	57	20
181	58	20
182	59	20
183	60	20
184	61	20
185	62	20
186	63	20
187	64	20
188	65	20
189	66	20
190	67	20
191	68	20
192	69	20
193	70	20
194	71	20
195	72	20
196	73	20
197	74	20
198	75	20
199	76	20
200	77	20
201	78	20
202	79	20
203	80	20
204	81	20
205	82	20
206	83	20
207	84	20

```
        85   20 /;


TABLE R(I,J) REQUEST LENGTH SUPPORTS OF I AND J

```


	P-A	P-B	P-C	P-D	H-A	H-B	C-A	C-B	I-A	I-B	B-A
1	0	0	0	0	0	0	0	0	15	0	0
2	0	0	16	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	13	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	13	0	0	0	0
6	0	0	0	0	0	0	0	14	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	15	0	0	0	0	0	0	0	0
10	0	0	14	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	14	0	0	0	0	0
13	13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	10	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	16	0	0
18	0	0	0	0	0	0	0	0	0	0	0
19	0	0	16	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	16	0	0	0	0	0	0	0	0
22	14	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	16	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0
26	0	15	0	0	0	0	0	0	0	0	0
27	11	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0
29	0	0	16	0	0	10	16	0	7	0	0
30	0	0	0	0	0	0	0	0	0	0	0
31	0	0	16	0	0	0	0	0	0	0	0
32	0	14	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	16	0	0	0	0	0
34	0	0	0	0	16	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	15	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0
38	0	0	12	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0
40	0	12	0	0	0	0	0	0	0	0	0

```
41     0    0    0    0    0   17    0    0    0    0    0
42     0    0    0    0    0    0    0    0    0    0   13
43     0    0   17    0    0    0    0    0    0    0    0
44     0    0    0    0    0    0    0    0    0    0    0
45     0    0    0    0    0    0   11    0    0    0    0
46     0   17    0    0    0    0    0    0    0    0    0
47     0    0   14    0    0    0    0    0    0    0    0
48     0    0    0    0    0    0    0    0    0    0    0
49     0    0    0    0    0    0    0    0    0    0   12
50     0    0    0    0    0    0    0    0    0    0    0
51     0    0    0    0    0   12    0    0    0    0    0
52     0    0   16    0    0    0    0    0    0    0    0
53     0    0    0    0    0    0    0    0    0    0    0
54     0    0    0    0    0    0   15    0    0    0    0
55     0    0    0    0    0    0    0    0    0    0    6
56     0   16    0    0    0    0    0    0   16    0    0
57     0    0    0    0    0   10    0    0    0    0    0
```



```
58    16    0    0    0    0    0    0    0    0    0    0
59     0    0   16    0    0    0    0    0    0    0    0
60     0    0    0    0    0    0    0    0    0    0   15
61     0    0    0    0    0    0    0    0    0    0    0
62     0    0    0    0    0    0    0    0    0    0    0
63     0    0   16    0    0    0    0    0    0    0    0
64    14    0    0    0    0    0    0    0    0    0    0
65     0    0    0    0    0    0    0    0    0    0    0
66     0    0    0    0    0    0    0    0    0    0    0
67     0    0   13    0    0    0    0    0    0    0    0
68     0   16    0    0    0    0    0    0    0    0    0
69    14    0    0    0    0    0    0    0    0    0    0
70     0    0    0    0    0    0    0    0    0    0    0
71     0    0    0    0    0    0    0    0    0    0    0
72     0    0   16    0    0    0    0    0    0    0    0
73     0    0    0    0    0    0    0    0    0    0   16
74     0    0    0    0    0    0    0    0    0    0    0
75     0    0    0    0    0   16    0    0    0    0    0
76     0    0    0    0   16    0    0    0    0    0    0
77     0    0   14    0    0    0    0    0    0    0    0
78    16    0    0    0    0    0    0    0    0    0    0
79     0   14    0    0    0    0    0    0    0    0    0
80     0    0    0    0    0    0    0    0    0    0   14
81     0    0    0    0    0    0    0    0    0    0    0
82     0    0    0    0    0    0    0   16    0    0    0
83     0    0    0    0    0    0    0    0    0    0    0
84     0    0    0    0    0   13    0    0    0    0    0
85     0    0    0    0   13    0    0    0    0    0    0
 +   B-B  L-A  L-B  G-A  G-B   PI  REF
```

300	1	0	0	0	0	0	0	0
301	2	0	0	0	0	0	0	0
302	3	16	0	0	0	0	0	0
303	4	0	0	0	0	0	0	0
304	5	0	0	0	0	0	0	0
305	6	0	0	0	0	0	0	0
306	7	0	0	0	0	0	0	12
307	8	0	0	0	15	0	0	0
308	9	16	0	0	0	0	14	12
309	10	0	0	0	0	0	0	0
310	11	0	0	0	0	0	16	0
311	12	0	0	0	0	0	0	0
312	13	0	0	0	0	0	0	0
313	14	12	0	0	0	0	0	0
314	15	0	0	0	0	0	0	0
315	16	0	0	0	14	0	0	0
316	17	0	0	0	0	0	0	0
317	18	0	0	0	0	0	0	14
318	19	0	0	0	0	0	0	0
319	20	0	0	14	0	0	0	0
320	21	0	0	0	0	0	0	0
321	22	0	0	0	0	0	0	0
322	23	0	0	0	0	0	0	0
323	24	0	0	0	0	0	15	0
324	25	0	0	13	0	0	0	0

325	26	0	0	0	0	0	0	0
326	27	0	0	0	0	0	0	0
327	28	0	0	13	0	0	0	0
328	29	0	11	0	0	0	13	0
329	30	0	0	11	0	0	0	0
330	31	0	0	0	0	0	0	0
331	32	0	0	0	0	0	0	0
332	33	0	0	0	0	0	0	0
333	34	0	0	0	0	0	0	0
334	35	15	0	0	0	0	0	0
335	36	0	0	0	0	0	0	0
336	37	0	0	0	14	0	0	0
337	38	0	0	0	0	0	0	0
338	39	0	0	16	0	0	0	0
339	40	0	0	0	0	0	0	0
340	41	0	0	0	0	0	0	0
341	42	0	0	0	0	0	0	0
342	43	0	0	0	0	0	0	0
343	44	0	0	0	0	0	14	0
344	45	0	0	0	0	0	0	0
345	46	0	0	0	0	0	0	0

```
 47      0    0    0    0    0    0    0
 48      0    0   16    0    0    0    0
 49      0    0    0    0    0    0    0
 50      0    0    0    0    0    0   16
 51      0    0    0    0    0    0    0
 52      0    0    0    0    0    0    0
 53      0    0    0   15    0    0    0
 54      0    0    0    0    0    0    0
 55      0    0    0    0    0   17    0
 56      0    0    0    0    0    0    0
 57      0    0    0    0    0    0    0
 58      0    0    0    0    0    0    0
 59      0    0    0    0    0    0    0
 60      0    0    0    0    0    0    0
 61      0    0    0    0    0    0   14
 62      0    0    0    0    0   15    0
 63     16    0    0    0    0    9    0
 64      0    0    0    0    0    0    0
 65      0    0    0    0    0    0   12
 66      0    0    0   15    0    0    0
 67      0    0    0    0    0    0    0
 68      0    0    0    0    0    0    0
 69      0    0    0    0    0    0    0
 70      0    0   13    0    0    0    0
 71     15    0    0    0    0    0    0
 72      0    0    0    0    0    0    0
 73      0    0    0    0    0    0    0
 74      0    0    0    0    0    0   14
 75      0    0    0    0    0    0    0
 76      0    0    0    0    0    0    0
 77      0    0    0    0    0    0    0
 78      0    0    0    0    0    0    0
 79      0    0    0    0    0    0    0
```



```
 80      0    0    0    0    0    0    0
 81     15    0    0    0    0    0    0
 82      0    0    0    0    0    0    0
 83      0   15    0    0    0    0    0
 84      0    0    0    0    0    0    0
 85      0    0    0    0    0    0    0
 ;

TABLE M(I,J) FEASIBLE SUPPORTS OF I AND J

        P-A  P-B  P-C  P-D  H-A  H-B  C-A  C-B  I-A  I-B  B-A
 1       0    0    0    0    0    0    0    0    1    0    0
 2       0    0    1    0    0    0    0    0    0    0    0
```

392	3	0	0	0	0	0	0	0	0	0	0	0
393	4	1	0	0	0	0	0	0	0	0	0	0
394	5	0	0	0	0	0	0	1	0	0	0	0
395	6	0	0	0	0	0	0	0	1	0	0	0
396	7	0	0	0	0	0	0	0	0	0	0	0
397	8	0	0	0	0	0	0	0	0	0	0	0
398	9	0	0	1	0	0	0	0	0	0	0	0
399	10	0	0	1	0	0	0	0	0	0	0	0
400	11	0	0	0	0	0	0	0	0	0	0	0
401	12	0	0	0	0	0	1	0	0	0	0	0
402	13	1	0	0	0	0	0	0	0	0	0	0
403	14	0	0	0	0	0	0	0	0	0	0	0
404	15	0	0	1	0	0	0	0	0	0	0	0
405	16	0	0	0	0	0	0	0	0	0	0	0
406	17	0	0	0	0	0	0	0	0	1	0	0
407	18	0	0	0	0	0	0	0	0	0	0	0
408	19	0	0	1	0	0	0	0	0	0	0	0
409	20	0	0	0	0	0	0	0	0	0	0	0
410	21	0	0	1	0	0	0	0	0	0	0	0
411	22	1	0	0	0	0	0	0	0	0	0	0
412	23	0	0	0	0	0	1	0	0	0	0	0
413	24	0	0	0	0	0	0	0	0	0	0	0
414	25	0	0	0	0	0	0	0	0	0	0	0
415	26	0	1	0	0	0	0	0	0	0	0	0
416	27	1	0	0	0	0	0	0	0	0	0	0
417	28	0	0	0	0	0	0	0	0	0	0	0
418	29	0	0	1	0	0	1	1	0	1	0	0
419	30	0	0	0	0	0	0	0	0	0	0	0
420	31	0	0	1	0	0	0	0	0	0	0	0
421	32	0	1	0	0	0	0	0	0	0	0	0
422	33	0	0	0	0	0	1	0	0	0	0	0
423	34	0	0	0	0	1	0	0	0	0	0	0
424	35	0	0	0	0	0	0	0	0	0	0	0
425	36	0	0	0	0	0	0	0	1	0	0	0
426	37	0	0	0	0	0	0	0	0	0	0	0
427	38	0	0	1	0	0	0	0	0	0	0	0
428	39	0	0	0	0	0	0	0	0	0	0	0
429	40	0	1	0	0	0	0	0	0	0	0	0
430	41	0	0	0	0	0	1	0	0	0	0	0
431	42	0	0	0	0	0	0	0	0	0	0	1
432	43	0	0	1	0	0	0	0	0	0	0	0

GAMS 2.20 VAX VMS 13-JAN-1993 13:40 PAGE 9
G E N E R A L A L G E B R A I C M O D E L I N G S Y S T E M
C O M P I L A T I O N

433	44	0	0	0	0	0	0	0	0	0	0	0
434	45	0	0	0	0	0	0	1	0	0	0	0
435	46	0	1	0	0	0	0	0	0	0	0	0
436	47	0	0	1	0	0	0	0	0	0	0	0
437	48	0	0	0	0	0	0	0	0	0	0	0

49	0	0	0	0	0	0	0	0	0	0	1
50	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	1	0	0	0	0	0
52	0	0	1	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	1	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	1
56	0	1	0	0	0	0	0	0	1	0	0
57	0	0	0	0	0	1	0	0	0	0	0
58	1	0	0	0	0	0	0	0	0	0	0
59	0	0	1	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	1
61	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0
63	0	0	1	0	0	0	0	0	0	0	0
64	1	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0
67	0	0	1	0	0	0	0	0	0	0	0
68	0	1	0	0	0	0	0	0	0	0	0
69	1	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	0	0	0	0	0
72	0	0	1	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	1
74	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	1	0	0	0	0	0
76	0	0	0	0	1	0	0	0	0	0	0
77	0	0	1	0	0	0	0	0	0	0	0
78	1	0	0	0	0	0	0	0	0	0	0
79	0	1	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	1
81	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	1	0	0	0
83	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	1	0	0	0	0	0
85	0	0	0	0	1	0	0	0	0	0	0

+	B-B	L-A	L-B	G-A	G-B	PI	REF
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1
8	0	0	0	1	0	0	0
9	1	0	0	0	0	1	1
10	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0

GAMS 2.20 VAX VMS 13-JAN-1993 13:40 PAGE 10
G E N E R A L A L G E B R A I C M O D E L I N G S Y S T E M

COMPILATION

487	12	0	0	0	0	0	0	0
488	13	0	0	0	0	0	0	0
489	14	1	0	0	0	0	0	0
490	15	0	0	0	0	0	0	0
491	16	0	0	0	1	0	0	0
492	17	0	0	0	0	0	0	0
493	18	0	0	0	0	0	0	1
494	19	0	0	0	0	0	0	0
495	20	0	0	1	0	0	0	0
496	21	0	0	0	0	0	0	0
497	22	0	0	0	0	0	0	0
498	23	0	0	0	0	0	0	0
499	24	0	0	0	0	0	1	0
500	25	0	0	1	0	0	0	0
501	26	0	0	0	0	0	0	0
502	27	0	0	0	0	0	0	0
503	28	0	0	1	0	0	0	0
504	29	0	1	0	0	0	1	0
505	30	0	0	1	0	0	0	0
506	31	0	0	0	0	0	0	0
507	32	0	0	0	0	0	0	0
508	33	0	0	0	0	0	0	0
509	34	0	0	0	0	0	0	0
510	35	1	0	0	0	0	0	0
511	36	0	0	0	0	0	0	0
512	37	0	0	0	1	0	0	0
513	38	0	0	0	0	0	0	0
514	39	0	0	1	0	0	0	0
515	40	0	0	0	0	0	0	0
516	41	0	0	0	0	0	0	0
517	42	0	0	0	0	0	0	0
518	43	0	0	0	0	0	0	0
519	44	0	0	0	0	0	1	0
520	45	0	0	0	0	0	0	0
521	46	0	0	0	0	0	0	0
522	47	0	0	0	0	0	0	0
523	48	0	0	1	0	0	0	0
524	49	0	0	0	0	0	0	0
525	50	0	0	0	0	0	0	1
526	51	0	0	0	0	0	0	0
527	52	0	0	0	0	0	0	0
528	53	0	0	0	1	0	0	0
529	54	0	0	0	0	0	0	0
530	55	0	0	0	0	0	1	0
531	56	0	0	0	0	0	0	0
532	57	0	0	0	0	0	0	0
533	58	0	0	0	0	0	0	0
534	59	0	0	0	0	0	0	0

```
   60      0      0      0      0      0      0      0
   61      0      0      0      0      0      0      1
   62      0      0      0      0      0      1      0
   63      1      0      0      0      0      1      0
   64      0      0      0      0      0      0      0
   65      0      0      0      0      0      0      1
GAMS 2.20 VAX VMS                              13-JAN-1993 13:40  PAGE     11
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
C O M P I L A T I O N


   66      0      0      0      1      0      0      0
   67      0      0      0      0      0      0      0
   68      0      0      0      0      0      0      0
   69      0      0      0      0      0      0      0
   70      0      0      1      0      0      0      0
   71      1      0      0      0      0      0      0
   72      0      0      0      0      0      0      0
   73      0      0      0      0      0      0      0
   74      0      0      0      0      0      0      1
   75      0      0      0      0      0      0      0
   76      0      0      0      0      0      0      0
   77      0      0      0      0      0      0      0
   78      0      0      0      0      0      0      0
   79      0      0      0      0      0      0      0
   80      0      0      0      0      0      0      0
   81      1      0      0      0      0      0      0
   82      0      0      0      0      0      0      0
   83      0      1      0      0      0      0      0
   84      0      0      0      0      0      0      0
   85      0      0      0      0      0      0      0
   ;

 TABLE BV(I,J) BEGINING OF  VISIBILITY

           P-A    P-B    P-C    P-D    H-A    H-B    C-A    C-B    I-A    I-B    B-A
   1        0      0      0      0      0      0      0      0     13      0      0
   2        0      0     26      0      0      0      0      0      0      0      0
   3        0      0      0      0      0      0      0      0      0      0      0
   4       51      0      0      0      0      0      0      0      0      0      0
   5        0      0      0      0      0      0     54      0      0      0      0
   6        0      0      0      0      0      0      0     54      0      0      0
   7        0      0      0      0      0      0      0      0      0      0      0
   8        0      0      0      0      0      0      0      0      0      0      0
   9        0      0    127      0      0      0      0      0      0      0      0
  10        0      0    138      0      0      0      0      0      0      0      0
  11        0      0      0      0      0      0      0      0      0      0      0
  12        0      0      0      0      0    139      0      0      0      0      0
  13      150      0      0      0      0      0      0      0      0      0      0
  14        0      0      0      0      0      0      0      0      0      0      0
  15        0      0    169      0      0      0      0      0      0      0      0
```

581	16	0	0	0	0	0	0	0	0	0	0	0
582	17	0	0	0	0	0	0	0	0	179	0	0
583	18	0	0	0	0	0	0	0	0	0	0	0
584	19	0	0	217	0	0	0	0	0	0	0	0
585	20	0	0	0	0	0	0	0	0	0	0	0
586	21	0	0	226	0	0	0	0	0	0	0	0
587	22	234	0	0	0	0	0	0	0	0	0	0
588	23	0	0	0	0	0	238	0	0	0	0	0
589	24	0	0	0	0	0	0	0	0	0	0	0
590	25	0	0	0	0	0	0	0	0	0	0	0
591	26	0	249	0	0	0	0	0	0	0	0	0
592	27	268	0	0	0	0	0	0	0	0	0	0
593	28	0	0	0	0	0	0	0	0	0	0	0
594	29	0	0	330	0	0	352	343	0	406	0	0

GAMS 2.20 VAX VMS 13-JAN-1993 13:40 PAGE 12

G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M

C O M P I L A T I O N

595	30	0	0	0	0	0	0	0	0	0	0	0
596	31	0	0	326	0	0	0	0	0	0	0	0
597	32	0	331	0	0	0	0	0	0	0	0	0
598	33	0	0	0	0	0	335	0	0	0	0	0
599	34	0	0	0	0	335	0	0	0	0	0	0
600	35	0	0	0	0	0	0	0	0	0	0	0
601	36	0	0	0	0	0	0	0	349	0	0	0
602	37	0	0	0	0	0	0	0	0	0	0	0
603	38	0	0	364	0	0	0	0	0	0	0	0
604	39	0	0	0	0	0	0	0	0	0	0	0
605	40	0	367	0	0	0	0	0	0	0	0	0
606	41	0	0	0	0	0	413	0	0	0	0	0
607	42	0	0	0	0	0	0	0	0	0	0	421
608	43	0	0	426	0	0	0	0	0	0	0	0
609	44	0	0	0	0	0	0	0	0	0	0	0
610	45	0	0	0	0	0	0	441	0	0	0	0
611	46	0	449	0	0	0	0	0	0	0	0	0
612	47	0	0	466	0	0	0	0	0	0	0	0
613	48	0	0	0	0	0	0	0	0	0	0	0
614	49	0	0	0	0	0	0	0	0	0	0	475
615	50	0	0	0	0	0	0	0	0	0	0	0
616	51	0	0	0	0	0	515	0	0	0	0	0
617	52	0	0	527	0	0	0	0	0	0	0	0
618	53	0	0	0	0	0	0	0	0	0	0	0
619	54	0	0	0	0	0	0	539	0	0	0	0
620	55	0	0	0	0	0	0	0	0	0	0	548
621	56	0	551	0	0	0	0	0	0	580	0	0
622	57	0	0	0	0	0	551	0	0	0	0	0
623	58	559	0	0	0	0	0	0	0	0	0	0
624	59	0	0	567	0	0	0	0	0	0	0	0
625	60	0	0	0	0	0	0	0	0	0	0	571
626	61	0	0	0	0	0	0	0	0	0	0	0

62	0	0	0	0	0	0	0	0	0	0	0
63	0	0	628	0	0	0	0	0	0	0	0
64	625	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0
67	0	0	638	0	0	0	0	0	0	0	0
68	0	653	0	0	0	0	0	0	0	0	0
69	660	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	0	0	0	0	0
72	0	0	669	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	670
74	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	736	0	0	0	0	0
76	0	0	0	0	736	0	0	0	0	0	0
77	0	0	739	0	0	0	0	0	0	0	0
78	756	0	0	0	0	0	0	0	0	0	0
79	0	757	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	769
81	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	792	0	0	0
83	0	0	0	0	0	0	0	0	0	0	0

84	0	0	0	0	0	807	0	0	0	0	0
85	0	0	0	0	807	0	0	0	0	0	0

+	B-B	L-A	L-B	G-A	G-B	PI	REF
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	39	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	81
8	0	0	0	108	0	0	0
9	117	0	0	0	0	119	161
10	0	0	0	0	0	0	0
11	0	0	0	0	0	138	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	163	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	173	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	193
19	0	0	0	0	0	0	0
20	0	0	225	0	0	0	0

673	21	0	0	0	0	0	0	0
674	22	0	0	0	0	0	0	0
675	23	0	0	0	0	0	0	0
676	24	0	0	0	0	0	238	0
677	25	0	0	244	0	0	0	0
678	26	0	0	0	0	0	0	0
679	27	0	0	0	0	0	0	0
680	28	0	0	271	0	0	0	0
681	29	0	324	0	0	0	342	0
682	30	0	0	325	0	0	0	0
683	31	0	0	0	0	0	0	0
684	32	0	0	0	0	0	0	0
685	33	0	0	0	0	0	0	0
686	34	0	0	0	0	0	0	0
687	35	340	0	0	0	0	0	0
688	36	0	0	0	0	0	0	0
689	37	0	0	0	355	0	0	0
690	38	0	0	0	0	0	0	0
691	39	0	0	365	0	0	0	0
692	40	0	0	0	0	0	0	0
693	41	0	0	0	0	0	0	0
694	42	0	0	0	0	0	0	0
695	43	0	0	0	0	0	0	0
696	44	0	0	0	0	0	440	0
697	45	0	0	0	0	0	0	0
698	46	0	0	0	0	0	0	0
699	47	0	0	0	0	0	0	0
700	48	0	0	466	0	0	0	0
701	49	0	0	0	0	0	0	0
702	50	0	0	0	0	0	0	480

703	51	0	0	0	0	0	0	0
704	52	0	0	0	0	0	0	0
705	53	0	0	0	535	0	0	0
706	54	0	0	0	0	0	0	0
707	55	0	0	0	0	0	539	0
708	56	0	0	0	0	0	0	0
709	57	0	0	0	0	0	0	0
710	58	0	0	0	0	0	0	0
711	59	0	0	0	0	0	0	0
712	60	0	0	0	0	0	0	0
713	61	0	0	0	0	0	0	582
714	62	0	0	0	0	0	614	0
715	63	619	0	0	0	0	625	0
716	64	0	0	0	0	0	0	0
717	65	0	0	0	0	0	0	632
718	66	0	0	0	636	0	0	0

```
67     0     0     0     0     0     0     0
68     0     0     0     0     0     0     0
69     0     0     0     0     0     0     0
70     0     0   665     0     0     0     0
71   668     0     0     0     0     0     0
72     0     0     0     0     0     0     0
73     0     0     0     0     0     0     0
74     0     0     0     0     0     0   728
75     0     0     0     0     0     0     0
76     0     0     0     0     0     0     0
77     0     0     0     0     0     0     0
78     0     0     0     0     0     0     0
79     0     0     0     0     0     0     0
80     0     0     0     0     0     0     0
81   771     0     0     0     0     0     0
82     0     0     0     0     0     0     0
83     0   806     0     0     0     0     0
84     0     0     0     0     0     0     0
85     0     0     0     0     0     0     0
;

TABLE EV(I,J) ENDING OF  VISIBILITY

        P-A   P-B   P-C   P-D   H-A   H-B   C-A   C-B   I-A   I-B   B-A
 1        0     0     0     0     0     0     0     0    28     0     0
 2        0     0    42     0     0     0     0     0     0     0     0
 3        0     0     0     0     0     0     0     0     0     0     0
 4       64     0     0     0     0     0     0     0     0     0     0
 5        0     0     0     0     0     0    67     0     0     0     0
 6        0     0     0     0     0     0     0    68     0     0     0
 7        0     0     0     0     0     0     0     0     0     0     0
 8        0     0     0     0     0     0     0     0     0     0     0
 9        0     0   142     0     0     0     0     0     0     0     0
10        0     0   152     0     0     0     0     0     0     0     0
11        0     0     0     0     0     0     0     0     0     0     0
12        0     0     0     0     0   153     0     0     0     0     0
13      163     0     0     0     0     0     0     0     0     0     0
14        0     0     0     0     0     0     0     0     0     0     0
```



```
15        0     0   179     0     0     0     0     0     0     0     0
16        0     0     0     0     0     0     0     0     0     0     0
17        0     0     0     0     0     0     0     0   195     0     0
18        0     0     0     0     0     0     0     0     0     0     0
19        0     0   233     0     0     0     0     0     0     0     0
20        0     0     0     0     0     0     0     0     0     0     0
21        0     0   242     0     0     0     0     0     0     0     0
22      248     0     0     0     0     0     0     0     0     0     0
```

765	23	0	0	0	0	0	254	0	0	0	0	0
766	24	0	0	0	0	0	0	0	0	0	0	0
767	25	0	0	0	0	0	0	0	0	0	0	0
768	26	0	264	0	0	0	0	0	0	0	0	0
769	27	279	0	0	0	0	0	0	0	0	0	0
770	28	0	0	0	0	0	0	0	0	0	0	0
771	29	0	0	346	0	0	362	359	0	413	0	0
772	30	0	0	0	0	0	0	0	0	0	0	0
773	31	0	0	342	0	0	0	0	0	0	0	0
774	32	0	345	0	0	0	0	0	0	0	0	0
775	33	0	0	0	0	0	351	0	0	0	0	0
776	34	0	0	0	0	351	0	0	0	0	0	0
777	35	0	0	0	0	0	0	0	0	0	0	0
778	36	0	0	0	0	0	0	0	364	0	0	0
779	37	0	0	0	0	0	0	0	0	0	0	0
780	38	0	0	376	0	0	0	0	0	0	0	0
781	39	0	0	0	0	0	0	0	0	0	0	0
782	40	0	379	0	0	0	0	0	0	0	0	0
783	41	0	0	0	0	0	430	0	0	0	0	0
784	42	0	0	0	0	0	0	0	0	0	0	434
785	43	0	0	443	0	0	0	0	0	0	0	0
786	44	0	0	0	0	0	0	0	0	0	0	0
787	45	0	0	0	0	0	0	452	0	0	0	0
788	46	0	466	0	0	0	0	0	0	0	0	0
789	47	0	0	480	0	0	0	0	0	0	0	0
790	48	0	0	0	0	0	0	0	0	0	0	0
791	49	0	0	0	0	0	0	0	0	0	0	487
792	50	0	0	0	0	0	0	0	0	0	0	0
793	51	0	0	0	0	0	527	0	0	0	0	0
794	52	0	0	543	0	0	0	0	0	0	0	0
795	53	0	0	0	0	0	0	0	0	0	0	0
796	54	0	0	0	0	0	0	554	0	0	0	0
797	55	0	0	0	0	0	0	0	0	0	0	554
798	56	0	567	0	0	0	0	0	0	596	0	0
799	57	0	0	0	0	0	561	0	0	0	0	0
800	58	575	0	0	0	0	0	0	0	0	0	0
801	59	0	0	583	0	0	0	0	0	0	0	0
802	60	0	0	0	0	0	0	0	0	0	0	586
803	61	0	0	0	0	0	0	0	0	0	0	0
804	62	0	0	0	0	0	0	0	0	0	0	0
805	63	0	0	644	0	0	0	0	0	0	0	0
806	64	639	0	0	0	0	0	0	0	0	0	0
807	65	0	0	0	0	0	0	0	0	0	0	0
808	66	0	0	0	0	0	0	0	0	0	0	0
809	67	0	0	651	0	0	0	0	0	0	0	0
810	68	0	669	0	0	0	0	0	0	0	0	0

69	674	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0
71	0	0	0	0	0	0	0	0	0	0	0
72	0	0	685	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0	0	0	686
74	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	752	0	0	0	0	0
76	0	0	0	0	752	0	0	0	0	0	0
77	0	0	753	0	0	0	0	0	0	0	0
78	772	0	0	0	0	0	0	0	0	0	0
79	0	771	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	783
81	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	808	0	0	0
83	0	0	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	820	0	0	0	0	0
85	0	0	0	0	820	0	0	0	0	0	0

+	B-B	L-A	L-B	G-A	G-B	PI	REF
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	55	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	93
8	0	0	0	123	0	0	0
9	133	0	0	0	0	133	173
10	0	0	0	0	0	0	0
11	0	0	0	0	0	154	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	175	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	187	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	207
19	0	0	0	0	0	0	0
20	0	0	239	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	253	0
25	0	0	257	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	0	0	284	0	0	0	0
29	0	335	0	0	0	355	0
30	0	0	336	0	0	0	0
31	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0

862	33	0	0	0	0	0	0	0
863	34	0	0	0	0	0	0	0
864	35	355	0	0	0	0	0	0

865	36	0	0	0	0	0	0	0
866	37	0	0	0	369	0	0	0
867	38	0	0	0	0	0	0	0
868	39	0	0	381	0	0	0	0
869	40	0	0	0	0	0	0	0
870	41	0	0	0	0	0	0	0
871	42	0	0	0	0	0	0	0
872	43	0	0	0	0	0	0	0
873	44	0	0	0	0	0	454	0
874	45	0	0	0	0	0	0	0
875	46	0	0	0	0	0	0	0
876	47	0	0	0	0	0	0	0
877	48	0	0	482	0	0	0	0
878	49	0	0	0	0	0	0	0
879	50	0	0	0	0	0	0	496
880	51	0	0	0	0	0	0	0
881	52	0	0	0	0	0	0	0
882	53	0	0	0	550	0	0	0
883	54	0	0	0	0	0	0	0
884	55	0	0	0	0	0	556	0
885	56	0	0	0	0	0	0	0
886	57	0	0	0	0	0	0	0
887	58	0	0	0	0	0	0	0
888	59	0	0	0	0	0	0	0
889	60	0	0	0	0	0	0	0
890	61	0	0	0	0	0	0	596
891	62	0	0	0	0	0	629	0
892	63	635	0	0	0	0	634	0
893	64	0	0	0	0	0	0	0
894	65	0	0	0	0	0	0	644
895	66	0	0	0	651	0	0	0
896	67	0	0	0	0	0	0	0
897	68	0	0	0	0	0	0	0
898	69	0	0	0	0	0	0	0
899	70	0	0	678	0	0	0	0
900	71	683	0	0	0	0	0	0
901	72	0	0	0	0	0	0	0
902	73	0	0	0	0	0	0	0
903	74	0	0	0	0	0	0	742
904	75	0	0	0	0	0	0	0
905	76	0	0	0	0	0	0	0
906	77	0	0	0	0	0	0	0
907	78	0	0	0	0	0	0	0

```
 79    0    0    0    0    0    0    0
 80    0    0    0    0    0    0    0
 81  786    0    0    0    0    C    0
 82    0    0    0    0    0    0    0
 83    0  821    0    0    0    0    0
 84    0    0    0    0    0    0    0
 85    0    0    C    0    0    0    0
 ;

SET D(I,H,J);
D(I,H,J)=YES$N(H,J)$N(I,J)$OFFDIAG(H,I);
GAMS 2.20 VAX VMS                                 13-JAN-1993 13:40  PAGE     18
G E N E R A L   A L G E B R A I C   M O D E L I N G   S Y S T E M
C O M P I L A T I O N


D(I,H,J)$((BV(I,J) GT EV(H,J) + TO(I)) OR(EV(I,J) +TO(H) LT BV(H,J)))
                                                                      =NO;
```

SRS Heuristic Code

The following Turbo Pascal program was written for the interchange and insertion procedures utilized in the SRS heuristic. The major procedures in this program are:

- Reads in the SRS low altitude and medium/high altitude databases
- Reads in the results of each MIP block
- Performs a two satellite support interchange procedure
- Performs a medium/high altitude support insertion procedure
- Performs a three satellite support interchange procedure

The Turbo Pascal code for the program is shown below:

```
program SRS_PROG;
{    Author: Capt Tim Gooley                                            }
{                                                                       }
{    Program Description:  This program was part of an overall         }
```

```
{    thesis effort to automate the intial 24 hour schedule for the     }
{    the Satellite Range Scheduling Process.                           }
{    This program does the following:                                  }
{                                                                      }
{      1.  Reads in the SRS low altitude and medium/high altitude      }
{          databases                                                   }
{      2.  Reads in the results of each MIP block                      }
{      3.  Performs a low altitude support interchange procedure       }
{      4.  Performs a medium/high altitude support insertion procedure }
{      5.  Performs a medium/high altitude support interchange procedure}
{                                                                      }
{                                                                      }
{  ARRAY DEFINITIONS                                                   }
{                                                                      }
{  sch:  array of scheduled low altitude supports  determined by       }
{        GAMS MIP program.  These sup/GTS combos will be read into     }
{        schd array for scheduled supports                             }
{  sup:  array containing the satellite support # and all GTS that     }
{        are visible to that GTS.  data shows that there are not more }
{        12 GTS visible that a support is visible at.                  }
{        EX:  SUP[10,2]= 7 means that the 10th support is visible to  }
{             GTS 7, SUP[10,3]=12 -10th support also visible to GTS 12}
{ bvis:  array containing the beginning of visibility for GTS that a  }
{        support is visible to.                                        }
{ evis:  array containing the ending of visibility for GTS that a      }
{        support is visible to.                                        }
{ schd:  array containing supports that are scheduled                  }
{        SCH[I,1] - support # that is scheduled                        }
{        SCH[I,2] - GTS support is scheduled at                        }
{        SCH[I,3] - start time for support                             }
{        SCH[I,4] - turnaround time for support                        }
{        SCH[I,5] - request length of the support                      }
{unschd: array of unscheduled supports at current time                 }
{turn  :  array of turnaround times for all supports                   }
{ wgt  :  array that holds the number of GTSs a support is visible to }
{space :  array that will hold the schedule for an entire day for a   }
{         GTS.  this array is built every time a support is considered}
{         to be scheduled at a GTS                                     }
{hold  : in low flier 2-opt interchnage holds unscd support info       }
{         HOLD[I,1] - GTSs unshcd support is visible to (hold[i,1])   }
{         HOLD[I,2] -  the sup currently at GTS at that time           }
{         HOLD[I,3] - START TIME of unschd support at  that GTS        }
{         HOLD[I,4] - TURN of unschd support at that GTS               }
{         HOLD[I,5] - request length  of unschd support                }


  TYPE
      MATSV= ARRAY [1..420,1..15] OF INTEGER;   {420}
      MATSC= ARRAY [1..85,1..18] OF INTEGER;   {75,18}
      MATTW= ARRAY[1..420] OF INTEGER;          {420}
```

```
MATHO= ARRAY[1..15,1..5] OF INTEGER;
MATSD= ARRAY[1..400,1..6] OF INTEGER;      {400}
MATUS= ARRAY[1..300] OF INTEGER;           {300}
MATSP= ARRAY[1..1440] OF INTEGER;
AGRND = ARRAY [1..18] OF STRING[6];
GRND= STRING[6];
INIT12= ARRAY[1..14] OF INTEGER;
MPOS = ARRAY[1..300] OF INTEGER;           {300}
MFREE = ARRAY[1..18] OF INTEGER;
MATLF = ARRAY[1..30] OF INTEGER;
MAXSP   = ARRAY[1..18, 1..2] OF INTEGER;
MATD    =  ARRAY[1..18] OF INTEGER;


{SRS DATA BASE RECORD                                                }
{                                                                    }
{SRSFILE,SRSRED  - FILENAME IN ASSIGN STAT, AND REC                  }
{SUP      - HOLDS SUPPORT #                                          }
{GTS      - GTS NAME                                                 }
{BV       - BEGINING VISIBILITY                                      }
{EV       - ENDING VISIBILITY                                        }
{REQ      - REQUEST LENGTH OF SUPPORT                                }
{TOT      - TURNAROUND TIME FOR SUPPORT                              }
{ID       - IRON NO                                                  }
{REV      -  REV NO                                                  }
{BSRT     -  BEGINNING OF STARTING TIME                              }
{ESRT     -  END OF STARTING TIME                                    }
{SCLD     - INDICATES IF SUPPORT IS SCHEDULED AT A GTS               }
{           S - THE SUPPORT IS SCHEDULED                             }
{           N - THE SUPPORT IS NOT SCHEDULED                         }


VAR
  MATCH,OK,GO,ISERT,GOON,STOP : BOOLEAN;

VAR
     sch : MATSC;        {75*18*2= 2700}
     bvis: MATSV;        {425*15*2= 10800}
     evis: MATSV;        {425*12*2= 10800}
     schd: MATSD;        {400*5*2=  4000}
   uschd : MATUS;        {300*2= 600}
    TURN : MATTW;        {425*2=900}
     WGT : MATTW;         {425*2=900}
      SUP: MATSV;         {425*12*2 = 10800}
     HOLD: MATHO;         {15*5*2=150}
     SGTS: AGRND;         {18*6=108}
     POS:  MPOS;          {300*2 = 600}
     FREE: MFREE;          {18*2 =36}
   LF_USCH :MATLF;           {30*2=60}
   FRE_SPC :MAXSP;
```

```
      SPACE   :MATSP;
      DOWN1    :MATD;


 {    INDEX VARIABLES - COUNTERS                                     }

        I,j,K,L,M,N,NUM,FLG,LAST: integer;

{  TSU- HOLDS TEMPORARY SUPPORT, BEG VIS, END VIS,REQ               }

     TSUP,TSU,BEG,ENDD,REQ,TUR,NUMREC,FLAG   :INTEGER;
     USUP,SUPP,UTUR,UBEG,UGTS,UREQ,GTS,UNUM :INTEGER;

{   MAX, MIN - SPACE FOR AN INSERTION OF A SUPPORT TO GTS          }
{  MINL, MAXL - TIME WITHIN DAY FOR SUPPORT TO BEGIN  ·            }

     MAX,MIN,MAXL,MINL,TMAX,TMIN,TMAXL,TMINL,MINGTS,MAXGTS  :INTEGER;

{OPT VARIABLF, USED IN 3 OPT INTERCHANGE FOR HF                    }

     OPTSUP,OPTGTS,OPTLOC,OPTLEN,OPTDIFF   :INTEGER;
     INT1_LOC,INT1_SUP,INT1_GTS,INT1_POS   :INTEGER;
     INT2_LOC,INT2_SUP,INT2_GTS,INT2_POS   :INTEGER;
     USH_LOC,USH_GTS,TOTHF,POSN,HFNUM,TDOWN :INTEGER;
     AVGHF  :REAL;

     LFSCHD,LFUSCHD,HFTOT,SUPLOC,GTSNUM,USCH  :INTEGER;
     USCHD_NUM,POS_NUM,FLEX                   :INTEGER;
     RATIO  : REAL;

{TEXT FILE VARIABLES                                               }
{                                                                  }
{INFILE, INFILE1  - TEXT FILENAMES                                 }
{ SU       - HOLDS SUPPORT #                                       }
{TGTS      - GTS NAME                                              }
{ABV,AEV  - HOLDS BEGIN AND ENDING VISIBILITY                      }
{ID,TA,RE - IRON NO, TURNAROUND TIME AND REQUEST LEN               }
{REVV    -   REV NO                                                }


        INFILE :TEXT;
        INFILE1 :TEXT;
        INFILELK :TEXT;
         ABV,AEV,TA,RE,ID,SU : INTEGER;
         REVV : REAL;
         TGTS: GRND;
         CH: STRING[1];
```

```
(***************************************************************************)
(*                                                                         *)
(*   Initialization procedure to zero out all arrays or to set to          *)
(*   constant value                                                        *)
(*                                                                         *)
(***************************************************************************)

PROCEDURE INIT ;

 VAR I,J: INTEGER;

 BEGIN
      SGTS[1]:='POGO-A';
      SGTS[2]:='POGO-B';
      SGTS[3]:='POGO-C';
      SGTS[4]:='HULA-A';
      SGTS[5]:='HULA-B';
      SGTS[6]:='COOK-A';
      SGTS[7]:='COOK-B';
      SGTS[8]:='INDI-A';
      SGTS[9]:='BOSS-A';
      SGTS[10]:='BOSS-B';
      SGTS[11]:='LION-A';
      SGTS[12]:='LION-B';
      SGTS[13]:='GUAM-A';
      SGTS[14]:='GUAM-B';
      SGTS[15]:='PIKE-A';
      SGTS[16]:='REEF-A';
      FOR I:= 1 TO 420 DO
      FOR J:= 1 TO 15 DO
       BEGIN
        BVIS[I,J]:=0;
        EVIS[I,J]:=0;
        SUP[I,J]:=0;
        TURN[I]:=0;
        WGT[I]:=0;
       END;

      FOR I:= 1 TO 85 DO
      FOR J:= 1 TO 18 DO
       BEGIN
        SCH[I,J]:=0;
       END;

       FOR I:= 1 TO 400 DO
       FOR J:= 1 TO 6 DO
        BEGIN
          SCHD[I,J]:=0;
        END;

       FOR I:= 1 TO 300 DO
```

```
        BEGIN
          USCHD[J]:=0;
          POS[J]:=0;
        END;
       FOR I:= 1 TO 15 DO
       FOR J:= 1 TO 5 DO
        BEGIN
          HOLD[I,J]:=0;
        END;
        FOR I:= 1 TO 18 DO
          DOWN1[I]:=0;

       END;  {PROCEDURE INIT}


(**************************************************************************)
(*                                                                        *)
(*        DOWN - reads in GTS downtime                                    *)
(*                                                                        *)
(**************************************************************************)

PROCEDURE DOWN(VAR TSCHD :MATSD; VAR DOWN1 :MATD; VAR TDOWN :INTEGER);

 VAR INFILED : TEXT;
      TGTS,SRTD,ENDT,I  :INTEGER;

BEGIN

     ASSIGN(INFILED,'C:\TP\SRSREAL\DOWN.DAT');
     {$I-} RESET(INFILED) {$I+};
     OK:= (IOResult = 0);

     IF OK=TRUE THEN
      BEGIN

       RESET(INFILED);
       I:=1;
       WHILE NOT  EOF(INFILED) DO
        BEGIN
          READLN(INFILED,TGTS,SRTD,ENDT);
          TSCHD[I,1]:=888;
          TSCHD[I,2]:=TGTS;
          TSCHD[I,3]:=SRTD;
          TSCHD[I,4]:=0;
          TSCHD[I,5]:=ENDT-SRTD;
          I:=I+1;
          DOWN1[TGTS]:=DOWN1[TGTS]+(ENDT-SRTD);
        END;
        TDOWN:=I;
        CLOSE(INFILED);
      END;
```

```
END;  {PROCEDURE DOWN1}




(***************************************************************************)
(*                                                                         *)
(*  The following procedures are used when the data from the data base    *)
(*  and the GAMS MIP programs are loaded into arrays and the PASCAL       *)
(*  randow access file.                                                    *)
(*                                                                         *)
(*  PRIMARY PURPOSE OF THESE PROCEDURES TO LOAD ARRAYS AND DATABASE       *)
(*                                                                         *)
(***************************************************************************)


{FSCHD   reads supports from SUP ASCII file and fills the SCHD array  }
{SCHD array holds all the supports that are scheduled                 }

PROCEDURE FSCHD(VAR INFILE1 :TEXT; SUP: MATSV; VAR TSCH : MATSC;
                                           VAR TSCHD:MATSD;
                                           VAR TUSCHD: MATUS);

VAR I,J,K,L,NUM, SU: INTEGER;
    GTS : STRING[6];
     CH : STRING[1];

BEGIN
          RESET(infile1);
           I:=0;
           WHILE NOT EOF(INFILE1) DO
            BEGIN
              I:=I+1;
              REPEAT
                 READLN(INFILE1,SU,NUM,CH,GTS);
              UNTIL (I=NUM) OR (EOF(INFILE1));
                 J:=0;
                 REPEAT
                  J:=J+1;
                 UNTIL SUP[J,1]=SU;
                 K:=0;
                 REPEAT
                   K:=K+1;
                 UNTIL (TSCH[I,K]= 1) OR (K =18);
                 IF K=18 THEN
                  BEGIN
                   IF TSCH[I,K]=0 THEN K:=19;
                  END;
```

```
                    CASE K OF
                    1..18: BEGIN
                             L:=0;
                             REPEAT
                              L:=L+1;
                             UNTIL TSCHD[L,1]=0;
                             M:=1;
                             REPEAT
                              M:=M+1;
                             UNTIL SUP[J,M]=K;
                             TSCHD[L,1]:=SU;
                             TSCHD[L,2]:=K;
                             TSCHD[L,3]:=BVIS[J,M];
                             TSCHD[L,4]:=TURN[J];
                             TSCHD[L,5]:=EVIS[J,M]-BVIS[J,M];
                             TSCHD[L,6]:=J;
                             M:=0;
                             REPEAT
                               M:=M+1;
                               IF TUSCHD[M]= SU THEN
                                BEGIN
                                 TUSCHD[M]:=999;
                                END;
                             UNTIL (TUSCHD[M]=0) OR (TUSCHD[M]=SU);
                           END;  {1..18 CASE}

                     19: BEGIN
                            L:=0;
                            REPEAT
                              L:=L+1;
                            UNTIL TUSCHD[L]=0;
                            TUSCHD[L]:=SU;
                           END;    {CASE = 19}
               END; {CASE STATEMENT}

               END;   { WHILE NOT EOF(INFILE)}

      END;  { FSCHD PROCEDURE}

    {FILLSCH   reads sch.dat ASCII file - fills the SCH array            }
    { SCH array is used to determine what supports are scheduled at a GTS}

     PROCEDURE FILLSCH(VAR INFILE,INFILE1: TEXT; VAR TSCH :MATSC);
      VAR I,J,NUM,ENDD,SU: INTEGER;
           GTS : STRING[6];
            CH : STRING[1];
      waste: string[80];

     BEGIN
            FOR I:= 1 TO 85 DO
            FOR J:=1 TO 18 DO
```

```
            BEGIN
             TSCH[I,J]:=0;
            END;
             RESET(INFILE1);
             WHILE NOT EOF(INFILE1) DO
              BEGIN
                READLN(INFILE1,SU,NUM,CH,GTS);
                ENDD:=NUM;
              END;
          CLOSE(INFILE1);
          RESET(infile);
          FOR I:= 1 to 5 do
           begin
            readln(infile,waste);
           end;
           FOR I := 1 to ENDD do
             BEGIN
               READLN(INFILE,NUM,Tsch[I,1],Tsch[I,2],Tsch[I,3],Tsch[I,4]
                ,Tsch[I,5],Tsch[I,6],Tsch[I,7],Tsch[I,8],Tsch[I,9],Tsch[I,10]);
             END;
           FOR I:= 1 to 3 do
            BEGIN
             readln(infile,waste);
            END;
           FOR I := 1 to ENDD do
             BEGIN
              READLN(INFILE,NUM,Tsch[I,11],Tsch[I,12],Tsch[I,13],Tsch[I,14]
              ,Tsch[I,15],Tsch[I,16],Tsch[I,17],Tsch[I,18]);
             END;
             close(infile);

 END;       {END FILLSCH PROCEDURE}


{ FILLARR  procedure that takes data from database and transfers the info}
{ to arrays SUP, BVIS,EVIS,TURN,WGT - all of these arrays hold the info  }
{used throughout the program                                            }

PROCEDURE FILLARR(TSU,TABV,TAEV,TTA,REE,FLG :INTEGER; TGTS: GRND; SGTS:AGRND;
                  VAR TSUP,TBVIS,TEVIS : MATSV;
                  VAR TTURN,TWGT :MATTW; VAR TUSCHD : MATUS);

 VAR I,J,NUM,K :INTEGER;
     MATCH   :BOOLEAN;

 BEGIN
           MATCH:=FALSE;
           I:=0;
           J:=1;
             REPEAT
               I:=I+1;
```

```
      IF TSUP[I,1]=TSU THEN MATCH:=TRUE;
    UNTIL (TSUP[I,1] =0) OR (TSUP[I,1] = TSU);
    NUM:=0;
    REPEAT
       NUM:=NUM+1;
    UNTIL SGTS[NUM]=TGTS;
    IF MATCH=TRUE THEN
     BEGIN
       REPEAT
       J:=J+1;
       UNTIL (TSUP[I,J]=0) OR (TSUP[I,J]=NUM) OR (J=15);
     END
   ELSE
     BEGIN
      J:=2;
      TSUP[I,1]:=TSU;
     END;
       IF TSUP[I,J]=0 THEN
         TWGT[I]:=TWGT[I]+1;
       TSUP[I,J]:=NUM;
       TBVIS[I,J]:=TABV;
       IF TAEV > 1440 THEN
         TAEV:=1440;
       TEVIS[I,J]:=TAEV;
       TTURN[I]:=TTA;

{ flg=1 indicates a high flier support                      }

       IF FLG=1 THEN
        BEGIN
           OK:=FALSE;
           K:=0;
           TBVIS[I,1]:=REE;
           REPEAT
             K:=K+1;
             IF TSU= TUSCHD[K] THEN
                 OK:=TRUE;
           UNTIL (TUSCHD[K]=0) OR (OK=TRUE);
           IF OK=FALSE THEN
             BEGIN
               TUSCHD[K]:=TSU;
               POS[K]:=I;
             END;  {OK=FALSE}
        END;  { IF FLG=1}
END;  {PROCEDURE FILLARR}


{  FILLS IN DATA BASE -for high fliers, a support that is visible to side A}
```

```
{ should also be visible to side B- this insures that a support is visible  }
{ to both sides.  This procedure insures that each support is visible to    }
{all the GTS it should be                                                   }


PROCEDURE HFILL(TUSCHD : MATUS; TPOS :MPOS; VAR TSUP :MATSV; VAR TWGT :MATTW;
                          VAR TBVIS,TEVIS : MATSV);

VAR I,J,K,L,A,NUM,FL1,FL2,FL3,FL4,REM,H,SUPNUM,NUMSUP  :INTEGER;
STOP : BOOLEAN;

BEGIN
          A:=0;
          REPEAT
          A:=A+1;
          UNTIL TUSCHD[A]=0;

          FOR L := 1 TO A DO
            BEGIN;
               I:=TPOS[L];
               J:=0;
               REPEAT
                 J:=J+1;
               UNTIL (TSUP[I,J]=0) OR (J=15);
               NUMSUP:=J-1;
               STOP:=FALSE;
            FOR H := 2 TO (J-1) DO
              BEGIN
                NUM:=TSUP[I,H];
                CASE NUM OF

                  1..4:  BEGIN
                          FL1:=0;
                          FL2:=0;
                          FL3:=0;
                          FL4:=0;
                          FOR K:= 2 TO J DO
                          BEGIN
                            CASE TSUP[I,K] OF
                             1:  FL1:=1;
                             2:  FL2:=1;
                             3:  FL3:=1;
                             4:  FL4:=1;
                            END;  {CASE TSUP}
                          END;  {FOR K = 2 ,J}

                            IF (FL4=0) AND (NUMSUP<15) AND (STOP=FALSE) THEN
                               BEGIN
                                NUMSUP:=NUMSUP+1;
                                TBVIS[I,NUMSUP]:=TBVIS[I,H];
                                TEVIS[I,NUMSUP]:=TEVIS[I,H];
```

```
         TWGT[I]:=TWGT[I]+1;
         TSUP[I,NUMSUP]:=4;
        END;
      IF (FL3=0) AND (NUMSUP<15) AND (STOP=FALSE) THEN
        BEGIN
         NUMSUP:=NUMSUP+1;
         TBVIS[I,NUMSUP]:=TBVIS[I,H];
         TEVIS[I,NUMSUP]:=TEVIS[I,H];
         TWGT[I]:=TWGT[I]+1;
         TSUP[I,NUMSUP]:=3;;
        END;
      IF (FL2=0) AND (NUMSUP<15) AND (STOP=FALSE) THEN
        BEGIN
         NUMSUP:=NUMSUP+1;
         TBVIS[I,NUMSUP]:=TBVIS[I,H];
         TEVIS[I,NUMSUP]:=TEVIS[I,H];
         TWGT[I]:=TWGT[I]+1;
         TSUP[I,NUMSUP]:=2;
        END;
      IF (FL1=0) AND (NUMSUP<15) AND (STOP=FALSE) THEN
        BEGIN
        NUMSUP:=NUMSUP+1;
         TBVIS[I,NUMSUP]:=TBVIS[I,H];
         TEVIS[I,NUMSUP]:=TEVIS[I,H];
         TWGT[I]:=TWGT[I]+1;
         TSUP[I,NUMSUP]:=1;
        END;
      IF (FL1=1) OR (FL2=1) OR (FL3=1) OR (FL4=1) THEN
        STOP:=TRUE;
    END; {CASE 1..4}

5..16:  BEGIN
         FL1:=0;
         REM:=NUM MOD 2;
         IF REM = 0 THEN
           BEGIN
             SUPNUM:=NUM-1;
             FOR K:= 2 TO J DO
              BEGIN
               IF TSUP[I,K]= SUPNUM THEN
                     FL1:=1;
               END; {K=1,J}
           END   {IF REM=0}
         ELSE
           BEGIN
             SUPNUM:=NUM+1;
             FOR K:= 2 TO J DO
              BEGIN
               IF TSUP[I,K]= SUPNUM THEN
                     FL1:=1;
               END; {K=1,J}
```

```
                          END;  {ELSE REM=0}

                          IF (FL1=0) AND (NUMSUP<15) THEN
                           BEGIN
                             NUMSUP:=NUMSUP+1;
                             TBVIS[I,NUMSUP]:=TBVIS[I,H];
                             TEVIS[I,NUMSUP]:=TEVIS[I,H];
                             TWGT[I]:=TWGT[I]+1;
                             TSUP[I,NUMSUP]:=SUPNUM;
                           END;  {FL=0 AND NUMSUP<15}
                        END;  {CASE 5-18}

                END;  {CASE NUM STATEMENT}

            END{FOR H=1 TO J}
           END;  {FOR L=}
END;  {PROCEDURE HFFILL}

(*************************************************************************)
(*                                                                       *)
(*  The following procedures are used during the LOW ALTITUDE 2-OPT      *)
(*  INTERCHANGE PROCEDURE.  determines flexibility of unscheduled support *)
(*  the overlapping schedu'led supports at each GTS, if a scheduled support*)
(*  can be rescheduled at another GTS                                    *)
(*                                                                       *)
(*************************************************************************)


{ BSORT  for an unscheduled support,  the scheduled supports are sorted  }
{at each of the GTS the Unsch support - the most flexibile Sch support is }
{ examined first for interchange                                          }

 PROCEDURE GTS_SORT(FIRST,LAST : INTEGER; VAR TRANK:INIT12;
                                        VAR THOLD: MATHO);
 VAR I,J,K, TEMP1: INTEGER;
     TEMP: ARRAY[1..5] OF INTEGER;
     SWITCH: BOOLEAN;

BEGIN
     FOR J:= 1 TO 5 DO
       TEMP[J]:=0;
     REPEAT
       SWITCH:=FALSE;
       FOR I:= FIRST TO LAST-1 DO
         BEGIN
           IF TRANK[I] < TRANK[I+1] THEN
             BEGIN
               TEMP1:=TRANK[I];
               TRANK[I]:=TRANK[I+1];
               TRANK[I+1]:=TEMP1;
```

```
                FOR J:= 1 TO 5 DO
                  BEGIN
                   TEMP[J]:=THOLD[I,J];
                   THOLD[I,J]:=THOLD[I+1,J];
                   THOLD[I+1,J]:=TEMP[J];
                  END;
                 SWITCH:=TRUE;
              END;  {IF TRANK}
           END;  {FOR I}
      UNTIL SWITCH=FALSE;

 END;  {PROCEDURE SUP_SORT}



{ SSORT  sorts the scheduled supports  by GTS   }

 PROCEDURE SSORT(FIRST,LAST : INTEGER; VAR TSCHD :MATSD);

 VAR I,J,K : INTEGER;
     TEMP: ARRAY[1..6] OF INTEGER;
     SWITCH: BOOLEAN;

BEGIN
     FOR J:= 1 TO 6 DO
       TEMP[J]:=0;
     REPEAT
       SWITCH:=FALSE;
       FOR I = FIRST TO LAST-1 DO
         BEGIN
           IF TSCHD[I,2] > TSCHD[I+1,2] THEN
             BEGIN
               FOR J:=1 TO 6 DO
                 BEGIN
                  TEMP[J]:=TSCHD[I,J];
                  TSCHD[I,J]:=TSCHD[I+1,J];
                  TSCHD[I+1,J]:=TEMP[J];
                 END;
               SWITCH:=TRUE;
             END;  {IF TRANK}
          END;  {FOR I}
     UNTIL SWITCH=FALSE;

 { SORT ARRAY FOR SAME GTS BUT BY INCREASING START TIME        }
     REPEAT
       SWITCH:=FALSE;
       FOR I:= 1 TO LAST DO
         BEGIN
           IF TSCHD[I,2]=TSCHD[I+1,2] THEN
             BEGIN
```

```
            IF TSCHD[I,3] > TSCHD[I+1,3] THEN
              BEGIN
               FOR J:=1 TO 6 DO
                 BEGIN
                  TEMP[J]:=TSCHD[I,J];
                  TSCHD[I,J]:=TSCHD[I+1,J];
                  TSCHD[I+1,J]:=TEMP[J];
                 END;
                 SWITCH:=TRUE;
              END;  {IF TSCHD>}
           END; {IF TEMP = }
        END;  {FOR I}
     UNTIL SWITCH=FALSE;


 END;  {PROCEDURE SSORT}



{ LSORT  sorts the UNSCHD  array from lowest non zero number to highest   }

 PROCEDURE LFSORT(LAST : INTEGER; VAR TUSCHD: MATUS; VAR TPOS :MPOS);

 VAR I,TEMP,TEMP1 : INTEGER;
     SWITCH: BOOLEAN;

BEGIN
     REPEAT
       SWITCH:=FALSE;
       FOR I:= 1 TO LAST-1 DO
         BEGIN
           IF TUSCHD[I] > TUSCHD[I+1] THEN
             BEGIN
               TEMP:=TUSCHD[I];
               TEMP1:=TPOS[I];
               TUSCHD[I]:=TUSCHD[I+1];
               TUSCHD[I+1]:=TEMP;
               TPOS[I+1]:=TEMP1;
               SWITCH:=TRUE;
             END;  {IF TUSCHD}
         END;  {FOR I}
     UNTIL SWITCH=FALSE;

 END;  {PROCEDURE LFSORT}


{PROCEDURE CHK - CHECKS TO SEE IF AN OVERLAP EXISTS BETWEEN A SUPPORT   }
{TRYING TO BE SCHEDULED AT A GTS AND IF THAT GTS HAS AVAILABLE SPACE FOR}
```

```
{ THE SUPPORT TO BE SCHEDULED                                              }

PROCEDURE CHK (BEG,ENDD,GTS,SU : INTEGER; TSCHD : MATSD; VAR MATCH: BOOLEAN);


VAR M,TBEG,TEND,FL : INTEGER;
    CK1,CK2,CK3 :  BOOLEAN;

BEGIN
           MATCH:=TRUE;
           FL:=0;
           M:=0;
{ DETERMINES OVERLAPPING SCHEDULED SUPPORT AT A GTS}

           REPEAT
             CK1:=FALSE;
             CK2:=FALSE;
             CK3:=FALSE;
             M:=M+1;
             IF (GTS=TSCHD[M,2]) AND (SU <> TSCHD[M,1]) THEN
              BEGIN
                 TBEG:=TSCHD[M,3]-TSCHD[M,4];
                 TEND:=TSCHD[M,3]+TSCHD[M,5];
                 IF (BEG > (TBEG-1)) AND (BEG < (TEND+1)) THEN
                   CK1:=TRUE;
                 IF (ENDD > (TBEG-1)) AND (ENDD < (TEND+1)) THEN
                    CK2:=TRUE;
                 IF (BEG < (TBEG+1)) AND ((ENDD+1) > TEND) THEN
                    CK3:=TRUE;
                  IF (CK1=TRUE) OR (CK2=TRUE) OR (CK3=TRUE) THEN
                   BEGIN
                     FL:=1;
                     MATCH:=FALSE;
                   END;   { IF OR }
              END;  { IF GTS =..}
           UNTIL (M=400) OR (SCHD[M,2]>GTS) OR (FL=1);


 END;  {PROCEDURE CHK}



{PROCEDURE SWITCH PLACES AN USCHEDULED SUPPORT THAT WAS ADDED TO   }
{SCHEDULE AS ARE RESULT OF THE INTERCHANGE PROCEDURE              }

PROCEDURE SWITCH(TSUP,GTS,BEG,TREQ,TTUR,TSU,UGTS,SBEG,STUR,
SREQ,TSUPLOC : INTEGER; VAR TSCHD : MATSD; VAR TUSCHD : MATUS);

VAR I,J,K : INTEGER;
```

```
BEGIN
      I:=0;
      REPEAT
       I:=I+1;
      UNTIL TSCHD[I,1]=0;
      TSCHD[I,1]:=TSU;
      TSCHD[I,2]:=UGTS;
      TSCHD[I,3]:=SBEG;
      TSCHD[I,4]:=STUR;
      TSCHD[I,5]:=SREQ;
      TSCHD[I,6]:=TSUPLOC;
      J:=0;
      REPEAT
      J:=J+1;
      UNTIL TUSCHD[J]=TSU;
      TUSCHD[J]:=999;
      I:=0;
      REPEAT
       I:=I+1;
      UNTIL TSCHD[I,1]=TSUP;
      TSCHD[I,1]:=TSUP;
      TSCHD[I,2]:=GTS;
      TSCHD[I,3]:=BEG;
      TSCHD[I,4]:=TTUR;
      TSCHD[I,5]:=TREQ;


END;   {PROCEDURE SWITCH}


{PROCEDURE SCHK - CHECKS TO SEE IF AN OVERLAP EXISTS BETWEEN A SUPPORT   }
{TRYING TO BE SCHEDULED AT A GTS AND IF THAT GTS HAS AVAILABLE SPACE FOR}
{ THE SUPPORT TO BE SCHEDULED                                            }

 PROCEDURE SCHK (BEG,ENDD,GTS : INTEGER: TSCHD : MATSD; VAR TSU :INTEGER;
                                                   VAR FL :INTEGER);
VAR  M,TBEG,TEND : INTEGER;
     MATCH,CK1,CK2,CK3 : BOOLEAN;


BEGIN
           MATCH:=FALSE;
           fl:=0;
           M:=0;
{ DETERMINES OVERLAPPING SCHEDULED SUPPORT AT A GTS}

           REPEAT
             CK1:=FALSE;
             CK2:=FALSE;
             CK3:=FALSE;
             M:=M+1;
```

```
            IF GTS=TSCHD[M,2] THEN
              BEGIN
                TBEG:=TSCHD[M,3]-TSCHD[M,4];
                TEND:=TSCHD[M,3]+TSCHD[M,5];
                IF (BEG > (TBEG-1)) AND (BEG < (TEND+1)) THEN
                  CK1:=TRUE;
                IF (ENDD > (TBEG-1)) AND (ENDD < (TEND+1)) THEN
                   CK2:=TRUE;
               IF (BEG < (TBEG+1)) AND ((ENDD+1) > TEND) THEN
                   CK3:=TRUE;
                 IF (CK1=TRUE) OR (CK2=TRUE) OR (CK3=TRUE) THEN
                  BEGIN
                    FL:=FL+1;
                    MATCH:=TRUE;
                    TSU:=TSCHD[M,1];
                  END;   { IF OR }
            END;  { IF THOLD[K,1]..}
          UNTIL (M=400) OR (TSCHD[M,2]>GTS);

 END;  {PROCEDURE SCHK}


{ FIND SUP - DETERMINES THE SCHEDULED SUPPORT THAT IS OVERLAPPING   }
{WITH THE UNSCHEDULED SUPPORT AT A SPECIFIC GTS, DETERMINES THE     }
{ FLEXIBILITY OF EACH SUPPORT - RANKS THE FLEXIBILITY OF SUPPORTS   }

PROCEDURE FIND_SUP(N: INTEGER; VAR THOLD : MATHO;
                                VAR TWGT,TTURN : MATTW;
                                VAR TSUP,TBVIS,TEVIS  : MATSV;
                                VAR FLAG,GTSNUM :INTEGER);



VAR    BEG, ENDD,K,M,I,J,TSU,FRST, LST: INTEGER;
       TBEG,TEND,GTS,DIFF  : INTEGER;
       RANK: INIT12;
       MATCH,OK : BOOLEAN;

BEGIN
       K:=1;
       WHILE THOLD[K,1] <>0 DO
         BEGIN
          BEG:=THOLD[K,3]-THOLD[K,4];
          ENDD:=THOLD[K,3]+THOLD[K,5];
          DIFF:=THOLD[K,5];
          IF DIFF > 8 THEN
           BEGIN
             GTS:=THOLD[K,1];
             SCHK(BEG,ENDD,GTS,SCHD,TSU,FLAG);
```

```
              THOLD[K,2]:=TSU;
              IF FLAG=0 THEN
                 GTSNUM:=K;
{DETERMINES THE FLEXIBILITY OF OVERLAPPINC SCHEDULED SUPPORT}

              L:=0;
              REPEAT
               L:=L+1;
              UNTIL TSUP[L,1]=TSU;
              RANK[K]:=TWGT[L];
            END;  {IF ENDD-BEG}
           K:=K+1;
         END; {WHILE THOLD}
         LST:=K;
         FRST:=1;
         IF K>2 THEN
           GTS_SORT(FRST,LST,RANK,THOLD);


END;  {PROCEDURE FIND_SUP}


{ INIT_HLD  initiliazes HOLD array}

 PROCEDURE INIT_HLD(VAR THOLD:MATHO);

 VAR I,J : INTEGER;

BEGIN
      FOR I:= 1 TO 15 DO
      FOR J:= 1 TO 5 DO
        THOLD[I,J]:=0;

END;  {PROCEDURE INIT_HLD}

{  USCH_GTS determines the GTS that an unscheduled support is visible to}
{  additionally, the beg and end of vis, turnaround time and request    }
{length are determined                                                   }

PROCEDURE USCH_GTS (TSU,FLAG,UNUM : INTEGER;  VAR THOLD : MATHO;
                                    VAR SUPN : INTEGER;
                                    VAR TSUP: MATSV);

 VAR M,K : INTEGER;

 BEGIN
              M:=M+1;
              SUPN:=0;
              IF FLAG=0 THEN
                 BEGIN
                   REPEAT
```

```
                SUPN:=SUPN+1;
                UNTIL SUP[SUPN,1]=TSU;
               END
           ELSE
               SUPN:=UNUM;
             K:=2;
          REPEAT
           THOLD[(K-1),1]:=TSUP[SUPN,K];
           THOLD[(K-1),3]:=BVIS[SUPN,K];
           THOLD[(K-1),4]:=TURN[SUPN];
           IF FLAG = 1 THEN
              THOLD[(K-1),5]:=BVIS[SUPN,1]
           ELSE
              THOLD[(K-1),5]:=EVIS[SUPN,K]-BVIS[SUPN,K];
             K:=K+1;
          UNTIL  (TSUP[SUPN,K]=0) OR (K=16);


 END;    {PROCEDURE USCH_GTS}


PROCEDURE LF_COLLECT( TDOWN1 :INTEGER; VAR TLFSCHD,TLFUSCHD :INTEGER;
            VAR TLF_USCH : MATLF; VAR TUSCHD : MATUS; VAR TPOS :MPOS);

VAR I,J,LAST,LST  :INTEGER;

BEGIN
           FOR I:= 1 TO 30 DO
             TLF_USCH[I]:=0;
           J:=0;
           REPEAT
           J:=J+1;
           UNTIL TUSCHD[J]=0;
           LAST:=J-1;
           LFSORT(LAST,TUSCHD,TPOS);
           J:=1;
           REPEAT
           TLF_USCH[J]:=TUSCHD[J];
           J:=J+1;
           UNTIL TUSCHD[J]=999;
           TLFUSCHD:=J-1;
           J:=0;
           REPEAT
           J:=J+1;
           UNTIL SCHD[J,1]=0;
           TLFSCHD:=J-1-TDOWN1;

END; {PROCEDURE LF_COLLECT}
```

```
{***********************************************************************}
{*                                                                     *}
{*    THE FOLLOWING PROCEDURES ARE USED IN THE HF INSERTION ROUTINES  *}
{*                                                                     *}
{***********************************************************************}


{ HFSORT determines the order in which high altitude satellite supports}
{ will be inserted into the schedule.  The order of priority will be : }
{ length of support request, flexibility,  visibility window          }

PROCEDURE HFSORT ( VAR TUSCHD : MATUS;  VAR TPOS : MPOS; VAR HFTOT :INTEGER);
VAR I,J,K,L,TEMP2,TEMP3   :INTEGER;
    TEMP       : ARRAY[1..300, 1..2] OF REAL;
    TEMP1      : ARRAY[1..2] OF REAL;
    temp4      :ARRAY[1..2] OF INTEGER;
  SWITCH       :BOOLEAN;

BEGIN
       J:=0;
       REPEAT
          J:=J+1
       UNTIL TUSCHD[J]=0;
       HFTOT:=J-1;
       FOR I:=1  TO J-1 DO
        BEGIN
            K:=POS[I];
            TEMP4[1]:=BVIS[K,1];
            TEMP4[2]:=EVIS[K,2]-BVIS[K,2];
            TEMP[I,1]:=TEMP4[2]/ TEMP4[1];
            TEMP[I,2]:=WGT[K];

          END; {FOR I :=1,J-1 }
          LAST:=J-2;

{ SORT TEMP ARRAY BY ORDER OF DECREASING REQUIREMENTS              }

     REPEAT
       SWITCH:=FALSE;
       FOR I:= 1 TO LAST DO
         BEGIN
           IF TEMP[I,1] > TEMP[I+1,1] THEN
             BEGIN
               FOR J:=1 TO 2 DO
                 BEGIN
                  TEMP1[J]:=TEMP[I,J];
                  TEMP[I,J]:=TEMP[I+1,J];
                  TEMP[I+1,J]:=TEMP1[J];
                 END;
```

```
              TEMP2:=TUSCHD[I];
              TUSCHD[I]:=TUSCHD[I+1];
              TUSCHD[I+1]:=TEMP2;
              TEMP3:=TPOS[I];
              TPOS[I]:=TPOS[I+1];
              TPOS[I+1]:=TEMP3;
            SWITCH:=TRUE;
          END;  {IF TEMP}
       END;  {FOR I}
    UNTIL SWITCH=FALSE;

{ SORT ARRAY FOR SAME REQUIRMENT VALUE BY DECREASING FLEXIBILITY      }
    REPEAT
      SWITCH:=FALSE;
      FOR I:= 1 TO LAST DO
        BEGIN
          IF TEMP[I,1]=TEMP[I+1,1] THEN
            BEGIN
             IF TEMP[I,2] > TEMP[I+1,2] THEN
               BEGIN
                 FOR J:=1 TO 3 DO
                   BEGIN
                     TEMP1[J]:=TEMP[I,J];
                     TEMP[I,J]:=TEMP[I+1,J];
                     TEMP[I+1,J]:=TEMP1[J];
                   END;
                    TEMP2:=TUSCHD[I];
                    TUSCHD[I]:=TUSCHD[I+1];
                    TUSCHD[I+1]:=TEMP2;
                    TEMP3:=TPOS[I];
                    TPOS[I]:=TPOS[I+1];
                    TPOS[I+1]:=TEMP3;
                    SWITCH:=TRUE;
               END;  {IF TEMP>}
            END; {IF TEMP = }
        END;  {FOR I}
    UNTIL SWITCH=FALSE;


END;  {PROCEDURE HFSORT}


{FREE_SPC procedure determines the amount of time available at a GTS that  }
{a satellite can be scheduled at.  This is absolute time, not useable time }
{ this is used as a guide to determine which GTS should be looked at first }
{ when inserting a support                                                 }

PROCEDURE FREE_SPC(TSCHD : MATSD; VAR TFREE: MFREE);

VAR I,J,DIFF,GTS  : INTEGER;
```

```
BEGIN
            FOR J := 1 TO 18 DO
              TFREE[J]:=1440;
            I:=0;
            REPEAT
             I:=I+1;
             GTS:=TSCHD[I,2];
             DIFF:=(TSCHD[I,3]+TSCHD[I,5]) - (TSCHD[I,3]-TSCHD[I,4]);
             TFREE[GTS]:=TFREE[GTS]-DIFF;
            UNTIL TSCHD[I,1]=0;


END;   {procedure free_spc}



{INS_GTS rank orders visible GTSs by most free space. that is the     }
{GTS  with the most free spacewill be the first one an               }
{ unscheduled support is attempted to be scheduled at                  }

PROCEDURE INS_GTS(TFREE : MFREE; VAR THOLD:MATHO);

VAR  I,J,K,L,TEMP2,NUM  : INTEGER;
    TEMP  : ARRAY [1..18,1..2] OF INTEGER;
    TEMP1 : ARRAY [1..2] OF INTEGER;
    TEMP3 : MFREE;
    TEMP4 : ARRAY[1..5] OF INTEGER;
    SWITCH,OK  : BOOLEAN;

BEGIN
      FOR I:= 1 TO 18 DO
         BEGIN
           TEMP[I,1]:=TFREE[I];
           TEMP[I,2]:=I;
         END;

     REPEAT
       SWITCH:=FALSE;
       FOR I:= 1 TO 17 DO
         BEGIN
           IF TEMP[I,1] < TEMP[I+1,1] THEN
             BEGIN
               FOR J:=1 TO 2 DO
                 BEGIN
                  TEMP1[J]:=TEMP[I,J];
                  TEMP[I,J]:=TEMP[I+1,J];
                  TEMP[I+1,J]:=TEMP1[J];
                 END;
               SWITCH:=TRUE;
```

```
            END;  {IF TEMP}
          END;  {FOR I}
      UNTIL SWITCH=FALSE;
      NUM:=0;
      FOR I:= 1 TO 18 DO
        BEGIN
          OK:=FALSE;
          K:=0;
          REPEAT
            K:=K+1;
            IF TEMP[I,2]=THOLD[K,1] THEN
              BEGIN
                OK:=TRUE;
                NUM:=NUM+1;
              END;  {IF TEMP=THOLD}
          UNTIL (OK=TRUE) OR (THOLD[K,1]=0);
           IF OK=TRUE THEN
             BEGIN
               FOR L:= 1 TO 5 DO
                BEGIN
                 TEMP4[L]:=THOLD[NUM,L];
                 THOLD[NUM,L]:=THOLD[K,L];
                 THOLD[K,L]:=TEMP4[L];
                END;  {FOR L = 1,5}
             END; {IF OK=TRUE}
        END; {FOR I = 1, 18}

END;  {PROCEDURE INS_GTS}

{INSERT  checks to see if a GTS has available space for the support to  }
{be inserted into the schedule, the procedure checks all openings and   }
{takes the locations that have the max and min open space               }
{the min open space has to be larger than the requirement               }

PROCEDURE INSERT (TBEG,TEND,TREQ :INTEGER; TSPACE :MATSP;
                  VAR TMAX,TMIN,TMINL,TMAXL :INTEGER; VAR TOK : BOOLEAN );

VAR      I,J,K,SP   :INTEGER;

BEGIN
         TMIN:=0;
         TMAX:=0;
         SP:=0;
         OK:=FALSE;
         IF TBEG <0 THEN
           TBEG:=0;
         FOR I:= TBEG TO TEND DO
           BEGIN
             IF (TSPACE[I] =0) AND (I< TEND) THEN
               SP:=SP+1
             ELSE
```

```
            BEGIN
             IF SP > (TREQ-1) THEN
              BEGIN
                 CASE TMAX OF

                 0:     BEGIN
                         TMAX:=SP;
                         TMAXL:=I-SP;
                        END;
            1..1400:     IF SP > TMAX THEN
                          BEGIN
                           TMAX:=SP;
                           TMAXL:=I-SP;
                         END;
                END;  {CASE TMAX}

                 CASE TMIN OF

                 0:     BEGIN
                         TMIN:=SP;
                         TMINL:=I-SP;
                        END;
            1..1400:     IF SP < TMIN THEN
                          BEGIN
                           TMIN:=SP;
                           TMINL:=I-SP;
                         END;
                END;  {CASE TMIN}
                SP:=0;
                OK:=TRUE;
              END { IF SP> TREQ}
            ELSE
              SP:=0;
          END;  { TSPACE =0}

        END; {FOR I=BEG TO END}

END;  {PROCEDURE INSERT}




{ FILLGTS  for a GTS and support combination fills the time a support }
{ needs to fulfill its support                                        }


PROCEDURE FILLGTS(SUPP,BSUP,ESUP :INTEGER; VAR TSPACE :MATSP);

VAR  I,J: INTEGER;
```

```
BEGIN
         FOR I:= BSUP TO ESUP DO
           BEGIN
            TSPACE[I]:=SUPP;
           END;
END;


{ FILL_SPC  for a specific GTS determines the scheduled supports #,        }
{the beg (end)  of the support requirement start time - turn,(start + req)}

PROCEDURE FILL_SPC (GT : INTEGER; TSCHD : MATSD; VAR TSPACE : MATSP);

VAR I,J,K,M,N,TBEG, TENDD,TSUPP : INTEGER;

BEGIN

       M:=1;
      REPEAT
        IF GT=TSCHD[M,2] THEN
         BEGIN
          TSUPP:=TSCHD[M,1];
          TBEG:=TSCHD[M,3]-TSCHD[M,4];
          TENDD:=TSCHD[M,3]+TSCHD[M,5];
          FILLGTS(TSUPP,TBEG,TENDD,TSPACE);
        END;
         M:=M+1;
      UNTIL (TSCHD[M,2]=0) OR (M=-00) OR (TSCHD[M,2] >GT);


 END;   { PROCEDURE FILL_SPC}

{  Initializes the space array                                        }

PROCEDURE INT_SPC(VAR TSPACE : MATSP);

VAR I:INTEGER;

 BEGIN
    FOR I:= 1 TO 1440 DO
     TSPACE[I]:=0;

END;   {PROCEDURE INIT_SPC}


{CHK_INS  checks to see if the openings at a GTS are the best location  }
{ a simple rule of thumb is applied: if the min open space - req time<5 }
{or max open space - req > 60 then this location is taken as satisfactory}

PROCEDURE CHK_INS(REQ,GTS,TMAX,TMIN,TMAXL,TMINL : INTEGER;
```

```
    VAR MAX,MIN,MAXL,MINL,MINGTS,MAXGTS :INTEGER; VAR TMATCH : BOOLEAN);

BEGIN
                    IF TMAX-REQ > 59 THEN
                      BEGIN
                        MAX:=TMAX;
                        MAXL:=TMAXL;
                        TMATCH:=TRUE;
                        IF MIN=1000 THEN
                          MIN:=999;
                      END;
                     IF TMIN-REQ <6 THEN
                       BEGIN
                         MIN:=TMIN;
                         MINL:=TMINL;
                         TMATCH:=TRUE;
                        IF MAX=0 THEN
                          MAX:=1;

                       END;
                    IF TMATCH=FALSE THEN
                      BEGIN
                        IF MIN > TMIN THEN
                          BEGIN
                            MIN:=TMIN;
                            MINGTS:=GTS;
                            MINL:=TMINL;
                          END;
                        IF MAX < TMAX THEN
                          BEGIN
                            MAX:=TMAX;
                            MAXGTS:=GTS;
                            MAXL:=TMAXL;
                          END;
                      END;  {IF MATCH=FALSE AND TMIN<>TMAX}
END; {PROCEDURE CHK_INS}

{ ADDSUP adds the inserted high flier support into the SCHD array     }
{additionally this procedure also updates the free space array for the}
{GTS the support was scheduled to                                     }


PROCEDURE ADDSUP (TSUP,TGTS,TBEG,TTUR,TREQ, TNUM,TSCH,TPOS_NUM : INTEGER;
                   VAR TSCHD :MATSD; VAR TFREE :MFREE; VAR TUSCHD :MATUS );

VAR DIFF :INTEGER;

BEGIN
          TSCHD[TSCH,1]:=TSUP;
          TSCHD[TSCH,2]:=TGTS;
          TSCHD[TSCH,3]:=TBEG;
```

```
        TSCHD[TSCH,4]:=TTUR;
        TSCHD[TSCH,5]:=REQ;
        TSCHD[TSCH,6]:=TPOS_NUM;
        TUSCHD[TNUM]:=999;
        DIFF:=(TBEG+REQ)-(TBEG-TTUR);
        TFREE[TGTS]:=TFREE[TGTS]-DIFF;

END; {PROCEDURE ADDSUP}



{INS_CK If a insertion is possible, this determines the best location    }
{ and GTS for the match                                                  }
PROCEDURE INS_CK (TMAX,TMIN,TREQ,TMAXL,TMINL,TMAXGTS,TMINGTS,NUM :INTEGER;
                  TMATCH: BOOLEAN; VAR TTGTS,TBEG : INTEGER);

 VAR  TTMIN,TTMAX :INTEGER;

BEGIN
             IF MATCH =TRUE THEN
               BEGIN
                 IF TMAX-TREQ >59 THEN
                   TBEG:=TMAXL+TURN[NUM];
                 IF TMIN-TREQ <6 THEN
                   TBEG:=TMINL+TURN[NUM];
               END
             ELSE
               BEGIN
                  IF TMAX-TREQ >29 THEN
                     TTMAX:=TMAX-TREQ-30
                  ELSE
                     TTMAX:=TMAX-TREQ;
                 IF TMIN-TREQ >29 THEN
                     TTMIN:=TMIN-TREQ-30
                  ELSE
                     TTMIN:=TMIN-TREQ;
                  IF TTMAX > TTMIN THEN
                    BEGIN
                      TBEG:=TMINL+TURN[NUM];
                      TTGTS:=TMINGTS
                    END
                ELSE
                   BEGIN
                     TBEG:=TMAXL+TURN[NUM];
                     TTGTS:=TMAXGTS
                   END;
               END;  {ELSE MATCH = TRUE}
 END; {PROCEDURE INS_CK}

{*************************************** ********************************}
{*                                                                      *}
```

```
{*          BEGINNING OF PROCEDURES FOR HF 3OPT INTERCHANGE             *}
{*                                                                      *}
{**********************************************************************}


{ FIND_GTS determines if the GTS in Schd array is one of the  GTS    }
{the unscheduled support is visible from i.e. in the hold array      }
PROCEDURE FIND_GTS(M :INTEGER; TSCHD : MATSD; THOLD :MATHO;
                   VAR TGTS :INTEGER);
VAR J :INTEGER;

BEGIN
           TGTS:=19;
           J:=0;
           REPEAT
            J:=J+1;
            IF THOLD[J,1]= TSCHD[M,2] THEN
             TGTS:=TSCHD[M,2];
           UNTIL (THOLD[J,1]=TSCHD[M,2]) OR (THOLD[J,1]=0);
END; {PROCEDURE FIND_GTS}

{PARAM finds the beginning and ending visibility and request length for}
{the Support being added in 3 opt bumping                              }
PROCEDURE PARAM(TTPOS_NUM,TTGTS :INTEGER; VAR TTBEG,TTEND,TTREQ :INTEGER);

VAR N  :INTEGER;

BEGIN
           N:=1;
           REPEAT
             N:=N+1;
           UNTIL (SUP[TTPOS_NUM,N]=TTGTS) OR (N=15);
           TTBEG:=BVIS[TTPOS_NUM,N]-TURN[TTPOS_NUM];
           TTREQ:=BVIS[TTPOS_NUM,1]+TURN[TTPOS_NUM];
           TTEND:=EVIS[TTPOS_NUM,N];
END;  {PROCEDURE PARAM}


{SCDH_POS  determines the amount of free space, position in the day}
{and total length of free space plus the support of interest       }
PROCEDURE SCHD_POS(TTGTS, SPLOC,TTREQ :INTEGER; TSCHD : MATSD;
                   VAR TFIRST,TLAST,TDIFF,TSP :INTEGER);

VAR NLOC  :INTEGER;

BEGIN
               NLOC:=TSCHD[SPLOC+1,6];
               IF TTGTS<>TSCHD[SPLOC-1,2] THEN
                BEGIN
                 TFIRST:=0;
                 TLAST:=TSCHD[SPLOC,3]+TSCHD[SPLOC,5];
```

```
TDIFF:=TLAST;
TSP:=TSCHD[SPLOC,3]-TSCHD[SPLOC,4];
END;
IF TTGTS=TSCHD[SPLOC-1,2] THEN
BEGIN
TFIRST:=TSCHD[SPLOC-1,3]+TSCHD[SPLOC-1,5];
TLAST:=TSCHD[SPLOC,3]+TSCHD[SPLOC,5];
TDIFF:=TLAST-TFIRST;
TSP:=(TSCHD[SPLOC,3]-TSCHD[SPLOC,4])-TFIRST;
END;    {IF TGTS=TSCHD[M-1,2]}
IF TTGTS<>TSCHD[SPLOC+1,2] THEN
BEGIN
TFIRST:=TSCHD[SPLOC,3]-TSCHD[SPLOC,4];
TLAST:=1440;
TDIFF:=1440-TFIRST;
TSP:=1440-(TSCHD[SPLOC,3]+TSCHD[SPLOC,5]);
END;  {IF TGTS <>TSHCD[M+1,2]}

IF (TDIFF<TTREQ-1) AND (TSCHD[SPLOC+1,2]=TTGTS)
AND (BVIS[SPLOC,1]<>0)  AND (BVIS[NLOC,1]=0) THEN

BEGIN
TFIRST:=TSCHD[SPLOC,3]-TSCHD[SPLOC,4];
TLAST:=TSCHD[SPLOC+1,3]-TSCHD[SPLOC+1,4];
TDIFF:=TLAST-TFIRST;
TSP:=TLAST - (TSCHD[SPLOC,3]+TSCHD[SPLOC,5]);

END;    {IF TGTS=TSCHD[M+1,2]}

END; {PROCEDURE SCHD_POS}


{FIND_SPC determines the weighting factor to put on each candidate scheduled}
{support to determine if it will be interchanged.  The weight is the sum of }
{the max free time at each GTS this support is visible to - therefore the   }
{more GTSs a support is visible to the higher the weight as well as GTS that}
{have more free space will get a higher weight                              }

PROCEDURE FIND_SPC (TLOC :INTEGER; TSCHD: MATSD; VAR SP_WGT :INTEGER);


VAR     I,J,K,TSP,TFIRST,TLAST,TDIFF,TTGTS,TEMP1  :INTEGER;
        TMPGTS,TBEG,TEND,SPC,PLACE,CNT,AVG :INTEGER;
        TEMP: MAXSP;
        BEST :ARRAY[1..3] OF INTEGER;
        SWITCH  : BOOLEAN;

BEGIN
        TSP:=0;
        AVG:=0;
        SPC:=0;
```

```
TMPGTS:=0;
FOR I := 1 TO 18 DO
FOR J:= 1 TO 2 DO
 BEGIN
  TEMP[I,J]:=0;
 END;
FOR I:= 1 TO 3 DO
  BEST[I]:=0;
CNT:=1;
J:=2;
I:=1;
REPEAT
  TEMP[I,1]:=SUP[TLOC,J];
  J:=J+1;
  I:=I+1;
UNTIL (SUP[TLOC,J]=0) OR (J=15);
 REPEAT
     TTGTS:=TEMP[CNT,1];
      INT_SPC(SPACE);
      FILL_SPC(TTGTS,SCHD,SPACE);
      K:=1;
      REPEAT
        K:=K+1;
      UNTIL SUP[TLOC,K]=TTGTS;
      TBEG:=BVIS[TLOC,K]-TURN[TLOC];
      TEND:=EVIS[TLOC,K];
      IF TBEG < 0 THEN
         TBEG:=0;
     FOR I:= TBEG TO TEND DO
      BEGIN
       IF (SPACE[I] =0) AND (I< TEND) THEN
         SPC:=SPC+1
       ELSE
         BEGIN
          IF SPC > BEST[3] THEN
            BEGIN
             BEST[3]:=SPC;
             REPEAT
               SWITCH:=FALSE;
               FOR J:=1 TO 2 DO
                BEGIN
                 IF BEST [J] < BEST[J+1] THEN
                   BEGIN
                    TEMP1:=BEST[J];
                    BEST[J]:=BEST[J+1];
                    BEST[J+1]:=TEMP1;
                   END; {IF BEST<}
                   SWITCH:=TRUE;
                END;  {FOR J=}
              UNTIL SWITCH=FALSE;
            SPC:=0;
```

```
                  PLACE:=I;
                 END
               ELSE
                 SPC:=0;
              END;
            END;  {FOR I:= TBEG ,TEND}
         FOR J:= 1 TO 3 DO
          AVG:=AVG+BEST[J];
         TEMP[CNT,2]:=AVG DIV 3;
         AVG:=0;
         TMPGTS:=TTGTS;
         FOR J:= 1 TO 3 DO
         BEST[J]:=0;
         SPC:=0;
         CNT:=CNT+1;
       UNTIL TEMP[CNT,1]=0;
        SP_WGT:=0;
        FOR I:=1 TO 18 DO
         SP_WGT:=SP_WGT+TEMP[I,2]:

END;    {PROCEDURE FIND_SPC}


{PROCEED determines if the support meets the criteria for a candidate }
{support. criteria are: 1) be in visibility window, 2) not a low flier}
{and 3) not  GTS maintenance                                          }
PROCEDURE PROCEED(TFIRST,TLAST,TTBEG,TTEND,ITREQ,TLOC,SNUM :INTEGER;
                 TSCHD : MATSD; VAR TDIFF, TSP :INTEGER; VAR GOON :BOOLEAN);

VAR CHK1,CHK2,CHK3,CHK4  :BOOLEAN;
    START : INTEGER;

BEGIN
              CHK1:=FALSE;
              CHK2:=FALSE;
              CHK3:=FALSE;
              CHK4:=TRUE;
              IF (TFIRST >TTBEG-1) AND (TTEND-1 > TLAST) THEN
                CHK1:=TRUE;
              IF BVIS[TLOC,1]<>0 THEN       {INDICATES A LOW ALT SUPPORT}
                CHK2:=TRUE;
              IF TSCHD[SNUM,1]<>888 THEN   {888 INDICATES GTS DOWNTIME}
                CHK3:=TRUE;
              IF (TTEND-1 < TLAST) AND (TFIRST > TTBEG)
                                      AND (TFIRST <TTEND) THEN

                BEGIN
                  CHK1:=TRUE;
                  TDIFF:=TTEND-TFIRST;
                  START:=TSCHD[SNUM,3]-TSCHD[SNUM,4];
                  IF TTEND < START THEN
```

```
              TSP:=TTEND-TFIRST
            ELSE
              TSP:=START-TFIRST;
          END;
          START:=TSCHD[SNUM,3]-TSCHD[SNUM,4];
        IF (TTBEG-1<  START) AND (TFIRST < TTBEG-1)
              AND (TLAST < TTEND-1) THEN

          BEGIN
            CHK1.=TRUE;
            TDIFF:=TLAST-TTBEG;
            TSP:=START-TTBEG
          END;
        IF (TFIRST >TTEND-1) OR (TLAST < TTBEG-1) THEN
          CHK4:=FALSE;

        IF (TSP > TTREQ-1) AND (CHK4=TRUE) THEN
          BEGIN
           CHK1:=TRUE;
           CHK2:=TRUE;
           CHK3:=TRUE;
          END;
        IF (CHK1=TRUE) AND (CHK2=TRUE) AND (CHK3=TRUE) THEN
          GOON:=TRUE;

END;  {PROCEDURE PROCEED}


{ OPT_INS PROCEDURE DOES THE FOLLOWING                                  }
{  determines the location where the support length plus free space is  }
{  greater than the request length of the unscheduled support           }
{  the location chosen is based on the minimum request length of the    }
{  scheduled support                                                    }

PROCEDURE OPT_INS (TPOS_NUM :INTEGER; TSCHD :MATSD; THOLD :MATHO;
                   SPACE :MATSP VAR TMINL,TNSUP,TNGTS,BLOC :INTEGER;
                   VAR TOK,TMATCH : BOOLEAN );

VAR      I,J,K,SP,FIRST,LAST,DIFF,LOC,SCHLOC,SWGT,SREQ,SUPCNT  :INTEGER;
         TTGTS,TGTS,BEST,TBEG,TEND,TREQ,WIND,CNT,GTSCNT   :INTEGER;
         RAT,TRATIO  :REAL;
         CHK  :BOOLEAN;

BEGIN
          BEST:=0;
          TMINL:=0;
          SP:=0;
          TOK:=FALSE;
          K:=0;
          SUPCNT:=0;
          TRATIO:=0.0;
```

```
TTGTS:=0;
CNT:=0;
GTSCNT:=1;
 WHILE (GTSCNT<16) AND (THOLD[GTSCNT,1]<>0) DO
  GTSCNT:=GTSCNT+1;
 IF GTSCNT=16 THEN
   GTSCNT:=15;
 REPEAT
  CHK:=FALSE;
  SUPCNT:=SUPCNT+1;
  FIND_GTS(SUPCNT,TSCHD,THOLD,TGTS);
  IF (TGTS<>19) AND (TTGTS<>TGTS) THEN
    BEGIN
     CNT:=CNT+1;
     TTGTS:=TGTS;
     FIRST:=0;
     LAST:=1440;
    END;  {IF TGTS<>19 & TTGTS<>TGTS}
    PARAM(TPOS_NUM,TGTS,TBEG,TEND,TREQ);
     IF TBEG<0 THEN
      TBEG:=0;
  IF TGTS <> 19 THEN
    BEGIN

     LOC:=TSCHD[SUPCNT,6];
     SCHD_POS(TGTS,SUPCNT,TREQ,TSCHD,FIRST,LAST,DIFF,SP);
   PROCEED(FIRST,LAST,TBEG,TEND,TREQ,LOC,SUPCNT,TSCHD,DIFF,SP,CHK);
     IF (CHK=TRUE) THEN
      BEGIN

       IF (DIFF > TREQ-1) THEN
       BEGIN
         CASE SP-TREQ OF

       0..1000:       BEGIN
                       TOK:=TRUE;
                       TNSUP:=999;
                       TNGTS:=TGTS;
                       TMINL:=FIRST;
                       TMIN:=0;
                       BLOC:=0;
                       TMATCH:=TRUE;
                      END;  {CASE 0..1000}

     -1000..-1:    BEGIN
                     SWGT:=0;
                     FIND_SPC(LOC,TSCHD,SWGT);
                     SREQ:=BVIS[LOC,1]+TURN[LOC];
                     N:=1;
                     REPEAT
                       N:=N+1;
```

```
                                 UNTIL SUP[LOC,N]=TGTS;
                                 WIND:=EVIS[LOC,N]-BVIS[LOC,N];
                                 RAT:=SWGT*(WIND/SREQ);
                                 IF (RAT > TRATIO) THEN
                                   BEGIN
                                     TRATIO:=RAT;
                                     TOK:=TRUE;
                                     TNSUP:=TSCHD[SUPCNT,1];
                                     TMINL:=FIRST;
                                     TNGTS:=TGTS;
                                     BLOC:=LOC;
                                     TMATCH:=FALSE;
                                   END;
                               END;  {CASE -1000..-1}

                               END;  {CASE STATEMENT}

                    END; { IF DIFF> TREQ}


               END; {IF CHK1& CHK2 & CHK3 =TRUE}
              END;  {IF TGTS <>19 WHICH IMPLIES TGTS=SCHD[M2,] }


          UNTIL (TMATCH=TRUE) OR (TSCHD[SUPCNT,1]=0) OR (SUPCNT=400)
                 OR (CNT>GTSCNT);

END;  {PROCEDURE OPT_INS}


{OPT_ADD if a free space is available for a support, OPT_ADD  adds the }
{unscheduled support where the interchnaged support used to be  and then}
{adds the bumped support into the location where the free space exceeds its}
{request length                                                            }
PROCEDURE OPT_ADD(TNUM,TUSUP,UGTS,ULOC,UPOS,SUP1,GTS1,LOC1,POS1,
                  SUP2,GTS2,LOC2,POS2 : INTEGER;
                  VAR TSCHD : MATSD;  TUSCHD :MATUS; TFREE : MFREE);
VAR I,J,K,L,M,N,DIFF    :INTEGER ;
    TBEG,TREQ,TTUR :INTEGER;


BEGIN
     L:=0;
     REPEAT
       L:=L+1;
     UNTIL TSCHD[L,1]=0;
     K:=1;
     REPEAT
      K:=K+1;
     UNTIL SUP[UPOS,K]=UGTS;
     TTUR:=TURN[UPOS];
```

```
      TREQ:=BVIS[UPOS,1];
      TBEG:=ULOC+TTUR;
      ADDSUP(TUSUP,UGTS,TBEG,TTUR,TREQ,TNUM,L,UPOS,TSCHD,TFREE,TUSCHD);
      IF SUP1<>999 THEN
       BEGIN
         I:=0;
         REPEAT
          I:=I+1;
         UNTIL TSCHD[I,1]=SUP1;
         M:=1;
         REPEAT
          M:=M+1;
         UNTIL SUP[POS1,M]=GTS1;
         DIFF:=(TSCHD[I,3]+TSCHD[I,5])-(TSCHD[I,3]-TSCHD[I,4]);
         TFREE[UGTS]:=TFREE[UGTS]+DIFF;
         TSCHD[I,2]:=GTS1;
         TSCHD[I,3]:=LOC1+TURN[POS1];
         TFREE[GTS1]:=TFREE[GTS1]-DIFF;
       END;  {SUP1 <>999}
      IF SUP2<>999 THEN
        BEGIN
          J:=0;
          REPEAT
           J:=J+1;
          UNTIL TSCHD[J,1]=SUP2;
          N:=1;
         REPEAT
          N:=N+1;
         UNTIL SUP[POS2,N]=GTS2;
         DIFF:=(TSCHD[J,3]+TSCHD[J,5])-(TSCHD[J,3]-TSCHD[J,4]);
         TFREE[GTS1]:=TFREE[GTS1]+DIFF;
         TSCHD[J,2]:=GTS2;
         TSCHD[J,3]:=LOC2+TURN[POS2];
         TFREE[GTS2]:=TFREE[GTS1]-DIFF;
       END;  {SUP2 <>999}

       SSORT(1,L,TSCHD);

END;  {PROCEDURE OPT-ADD}


{**************************************************************************}
{*                                                                        *}
{*       PROCEDURES THAT WRITE SCHEDULE RESULTS TO A FILE                 *}
{*                                                                        *}
{**************************************************************************}


{ INIT_FSPACE  initiliazes FRE_SPC array}
```

```
PROCEDURE INIT_FSPACE(VAR TFRE_SPC:MAXSP);

VAR I,J : INTEGER;

BEGIN
      FOR I:= 1 TO 18 DO
      FOR J:= 1 TO 2 DO
        TFRE_SPC[I,J]:=0;

END;  {PROCEDURE INIT_FSPACE}



{WRITES OUT ALL SCHEDULED SUPPORTS TO A FILE           }

PROCEDURE WRT_SCHD(TSCHD: MATSD; VAR INFILE1 :TEXT );

VAR SRT,TUR,REQ,GTSS,SUPP,I,J,K,L,M,SSUP,ID  :INTEGER;
     S10  :STRING[10];
     ST8  : STRING[8];
     ST50: STRING[50];
     TTGTS :STRING[6];
     CH: STRING[1];
     INFILEK: TEXT;
     SID : STRING[4];


BEGIN
      ASSIGN(INFILEK,'C:\TP\SRSREAL\LOOK.DAT');
      RESET(INFILEK);
      S10:='            ';
      ST8:='        ';
      CH:=' ';
      ST50:='Sup    ID       GTS    Srt Tme     Turn Tme       Req Len';
      M:=1;
      WRITELN(INFILE1,ST50);
      REPEAT
        SUPP:=TSCHD[M,1];
        GTSS:=TSCHD[M,2];
        SRT:=TSCHD[M,3];
        TUR:=TSCHD[M,4];
        REQ:=TSCHD[M,5];
        IF SUPP<>888 THEN
         BEGIN
          RESET(INFILEK);
          REPEAT
            READLN(INFILEK,SSUP,CH,TTGTS,CH,ID);
          UNTIL SUPP=SSUP;
          STR(ID:4,SID);
         END
```

```
        ELSE
          SID:='DOWN';
        WRITELN(INFILE1,SUPP:4,CH,SID,S10,SGTS[GTSS],ST8,SRT:4,ST8,TUR:4,ST8
                ,REQ:4);
        IF TSCHD[M,2]<>TSCHD[M+1,2] THEN
         WRITELN(INFILE1,' 000 down',s10,'NONE-A  0  0  0');
         M:=M+1;
      UNTIL TSCHD[M,1]=0;
      CLOSE(INFILE1);


END;  {PROCEDURE WRT_SCHD}

{WRT_USCHD Writes all the unscheduled supports to a file}
PROCEDURE WRT_USCHD( LF1 :MATLF; TUSCHD :MATUS; VAR INFILE1 :TEXT);

VAR I,J,K,L,LST,LAST,POSN :INTEGER;
    SPC    : STRING[50];

BEGIN
       SPC:='Uscheduled Low Flier and High Flier Supports';
       I:=0;
       REPEAT
         I:=I+1;
       UNTIL (LF1[I]=999) OR (LF1[I]=0);
       LAST:=I-1;
       J:=0;
       REPEAT
         J:=J+1;
       UNTIL (TUSCHD[J]=999) OR (TUSCHD[J]=0);
       LST:=J-1;
       WRITELN(INFILE1,SPC);
       SPC:='Low Flier Unscheduled Supports';
       WRITELN(INFILE1,SPC);
       IF LAST >1 THEN
        BEGIN
          FOR K:= 1 TO LAST DO
           BEGIN
            J:=0;
            REPEAT
             J:=J+1;
            UNTIL  (SUP[J,1]=LF1[K]) OR (J=420);
            WRITELN(INFILE1,SUP[J,1]:4,SUP[J,2]:4,SUP[J,3]:4,SUP[J,4]:4);
            WRITELN(INFILE1,BVIS[J,1]:4,BVIS[J,2]:4,BVIS[J,3]:4,BVIS[J,4]:4);
            WRITELN(INFILE1,EVIS[J,1]:4,EVIS[J,2]:4,EVIS[J,3]:4,EVIS[J,4]:4);
            WRITELN(INFILE1);
           END; {FOR K= 1,LAST}
        END; {IF LAST>1
          SPC:='High & Medium Flier Unscheduled Supports';
          WRITELN(INFILE1,SPC);
{          K=0;}
```

```
{          REPEAT}
{           K:=K+1;}
           FOR K:= 1 TO 300 DO
           BEGIN
           IF (TUSCHD[K] <>999) AND (TUSCHD[K]<>0) THEN
            BEGIN
             POSN:=POS[K];
             WRITELN(INFILE1,SUP[POSN,1]:4,SUP[POSN,2]:4,SUP[POSN,3]:4
                  ,SUP[POSN,4]:4,
                  SUP[POSN,5]:4,SUP[POSN,6]:4,SUP[POSN,7]:4,SUP[POSN,8]:4,
                  SUP[POSN,9]:4,SUP[POSN,10]:4,SUP[POSN,11]:4,SUP[POSN,12]:4,
                  SUP[POSN,13]:4,SUP[POSN,14]:4,SUP[POSN,15]:4);
             WRITELN(INFILE1);
             WRITELN(INFILE1,BVIS[POSN,1]:4,BVIS[POSN,2]:4,BVIS[POSN,3]:4
                ,BVIS[POSN,4]:4,
                BVIS[POSN,5]:4,BVIS[POSN,6]:4,BVIS[POSN,7]:4,BVIS[POSN,8]:4,
                BVIS[POSN,9]:4,BVIS[POSN,10]:4,BVIS[POSN,11]:4,
                BVIS[POSN,12]:4,BVIS[POSN,13]:4,BVIS[POSN,14]:4
                ,BVIS[POSN,15]:4);
             WRITELN(INFILE1);
             WRITELN(INFILE1,EVIS[POSN,1]:4,EVIS[POSN,2]:4,EVIS[POSN,3]:4
                  ,EVIS[POSN,4]:4,
                  EVIS[POSN,5]:4,EVIS[POSN,6]:4,EVIS[POSN,7]:4,EVIS[POSN,8]:4,
                  EVIS[POSN,9]:4,EVIS[POSN,10]:4,EVIS[POSN,11]:4,
                  EVIS[POSN,12]:4,EVIS[POSN,13]:4,EVIS[POSN,14]:4,
                  EVIS[POSN,15]:4);
             WRITELN(INFILE1);
             WRITELN(INFILE1);
           END;  {IF USCHD <>}
           END;  {FOR K = 1, 300}
{         UNTIL TUSCHD[K]=0;}
       CLOSE(INFILE1);

END;  {PROCEDURE WRT_USCHD}

{MAXSPC determines the max amount of free space for GTS  after schedule  }
{is built -  Hopefully less free space between supports indicates more   }
{supports scheduled                                                      }

PROCEDURE MAX_SPC (TSCHD: MATSD; VAR TFRE_SPC  :MAXSP);


VAR      I,J,K,SP,FIRST,LAST,TOT,TOT1,DIFF,TGTS,TOTAL,PLACE :INTEGER;

BEGIN
         SP:=0;
         DIFF:=0;
         M:=1;
         FOR I := 1 TO 18 DO
          BEGIN
           TFRE_SPC[I,1]:=0;
```

```
          TFRE_SPC[I,2]:=0;
         END;
        FOR TGTS:= 1 TO 18 DO
        BEGIN
         REPEAT
          IF TGTS=TSCHD[M,2] THEN
            BEGIN
             IF TGTS=SCHD[M+1,2] THEN
              BEGIN
               FIRST:=TSCHD[M,3]+TSCHD[M,5];
               LAST:=TSCHD[M+1,3]-TSCHD[M+1,4];
              END  {IF TTGTS=SCHD[M+1,2]}
             ELSE
              BEGIN
                FIRST:=TSCHD[M,3]+TSCHD[M,5];
                LAST:=1440;
              END; {ELSE IF TTGTS=SCHD[M+1,2]}
               SP:=LAST-FIRST;

               IF SP > DIFF THEN
                 BEGIN
                   DIFF:=SP;
                   PLACE:=FIRST;
                 END;

          END; {IF TTGTS=SCHD[M,2]}
           M:=M+1;
          UNTIL (SCHD[M,2]>TGTS) OR (SCHD[M,2]=0) OR (M=400);
           TFRE_SPC[TGTS,1]:=DIFF;
           TFRE_SPC[TGTS,2]:=PLACE;
           DIFF:=0;
        END;  {FOR TGTS = 1,18}

END;    {PROCEDURE MAX_SPC}


 PROCEDURE SCH_STATS(TLFSCHD,TLFUSCHD,THFTOT,TNUM :INTEGER; TFRE_SPC :MAXSP;
                     TFREE  :MFREE; TSGTS : AGRND);

 VAR INFILE :TEXT;
     SPC,SPC1,SP1    :STRING[80];
 LF_PER,HF_PER : REAL;
 HFSCHD,LFTOT,CNT        :INTEGER;
 PERC : ARRAY[1..18] OF REAL;
 SNUM: ARRAY[1..18] OF INTEGER;
 HIGH : ARRAY[1..100,1..2] OF INTEGER;

BEGIN
      FOR I:= 1 TO 100 DO
      FOR J:= 1 TO 2 DO
        HIGH[I,J]:=0;
```

```
FOR I:=1 TO 18 DO
  SNUM[I]:=0;
  I:=0;
REPEAT
  I:=I+1;
  CNT:=SCHD[I,2];
  IF CNT > 0 THEN
    SNUM[CNT]:=SNUM[CNT]+1;
  IF SCHD[I,1]<>888 THEN
    BEGIN
      J:=0;
      REPEAT
        J:=J+1;
      UNTIL (HIGH[J,1]=0) OR (HIGH[J,1]=SCHD[I,5]);
      IF HIGH[J,1] = 0 THEN
        HIGH[J,1]:=SCHD[I,5];
        HIGH[J,2]:=HIGH[J,2]+1;
    END;  {IF SCHD[I,1]<>888}
UNTIL SCHD[I,1]=0;

ASSIGN(INFILE,'C:\TP\SRSREAL\SCH-STAT.DAT');
REWRITE(INFILE);
SPC:='STATISTICS ON SCHEDULE';
WRITELN(INFILE,SPC);
SPC1:='                              ';
SP1:='     ';
WRITELN(INFILE,SPC);
SPC:='Low Flier Schedule Statistics';
WRITELN(INFILE,SPC);
WRITELN(INFILE,SPC1);
LF_PER:=TLFSCHD/(TLFSCHD+TLFUSCHD)*100;
LFTOT:=TLFSCHD+TLFUSCHD;
SPC:='TOT    TOT SCHD    PERCENT SCH';
WRITELN(INFILE,LFTOT,SP1,TLFSCHD,SP1,LF_PER);
HFSCHD:=THFTOT-TNUM;
HF_PER:=HFSCHD/THFTOT*100;
SPC:=' Med & High Flier Schedule Statistics';
WRITELN(INFILE,SPC);
WRITELN(INFILE,SPC1);
SPC:='TOT    TOT SCHD    PERCENT SCH';
WRITELN(INFILE,THFTOT,SP1,HFSCHD,SP1,HF_PER);
WRITELN(INFILE,SP1);
SPC:='Avg High Flier Density:';
WRITELN(INFILE,SPC);
WRITELN(INFILE,AVGHF:6:2);
WRITELN(INFILE,SP1);
SPC:='GTS STATISTICS';
WRITELN(INFILE,SPC);
SPC:=' GTS        % UTIL            NUM SUP    MAX FR SPC   DOWN TME';
WRITELN(INFILE,SPC);
FOR I:= 1 TO 18 DO
```

```
        BEGIN
         PERC[I]:=((1440-TFREE[I])/1440)*100;
         WRITELN(INFILE,TSGTS[I],SP1,PERC[I],SP1,SNUM[I],
                 SP1,TFRE_SPC[I,1],SP1,DOWN1[I]);
        END;
        WRITELN(INFILE,SP1);
       SPC:='HIGH FLIER SUPPORT STATISTICS';
       WRITELN(INFILE,SPC);
       SPC:=' Sup Len              No.';
       WRITELN(INFILE,SPC);
       J:=0;
       REPEAT
         J:=J+1;
         WRITELN(INFILE,HIGH[J,1]:6,HIGH[J,2]:4);
       UNTIL HIGH[J,1]=0;
       CLOSE(INFILE);

END;  {PROCEDURE SCHD_STATS}


{************************************************************************}
{*                                                                      *}
{*          BEGINNING OF MAIN PROGRAM                                   *}
{*                                                                      *}
{************************************************************************}

 BEGIN

      INIT;
       K:=0;

{Read in GTS down time for this day}
             DOWN(SCHD,DOWN1,TDOWN);

{assign data files                                       }


       ASSIGN(INFILELK,'c:\tp\srsREAL\LOOK.DAT');
       REWRITE(INFILELK);

(*data base file for all srs records                          *)

       ASSIGN(INFILE,'c:\tp\srsREAL\SUPLF1.DAT');
       RESET(INFILE);


{  Read data file with all srs records}
       NUMREC:=0;
       WHILE NOT EOF(INFILE) DO
        BEGIN
           READLN(INFILE,SU,CH,TGTS,ABV,AEV,RE,TA,ID,REVV);
```

```
      NUMREC:=NUMREC+1;
      FLAG:=0;  {INDICATES THAT THE SUPPORT IS A LOW FLIER}

      FILLARR(SU,ABV,AEV,TA,RE,FLAG,TGTS,SGTS,SUP,BVIS
              ,EVIS,TURN,WGT,USCHD);
      WRITELN(INFILELK,SU,CH,TGTS,CH,ID);


   END;
   close(infile);
   close(infilelk);
   FOR I:=1 TO 3 DO

    BEGIN

     CASE I OF

      1:  assign(infile,'C:\TP\SRSREAL\sch.dat');
      2:  ASSIGN(INFILE,'C:\TP\SRSREAL\SCH1.DAT');
      3:  ASSIGN(INFILE,'C:\TP\SRSREAL\SCH2.DAT');

     END;   {CASE}
      {$I-} RESET(INFILE) {$I+};
      OK:= (IOResult = 0);

      IF OK=TRUE THEN
        BEGIN



         CASE I OF

           1:  assign(infile1,'C:\TP\SRSREAL\sup.dat');
           2:  ASSIGN(INFILE1,'C:\TP\SRSREAL\sup1.DAT');
           3:  ASSIGN(INFILE1,'C:\TP\SRSREAL\Sup2.DAT');

          END;   {CASE}

          FILLSCH(INFILE,INFILE1,SCH);
          FSCHD(INFILE1,SUP,SCH,SCHD,USCHD);

        END;  {IF OK }

      END;  { FOR LOOP}

{ sort scheduled suppports by gts and start time }
       J:=0;
       REPEAT
        J:=J+1;
       UNTIL SCHD[J,2]=0;
       LAST:=J-1;
```

```
             SSORT(1,LAST,SCHD);


{sort unscheduled low altitude supports                          }
            L:=1;
            WHILE USCHD[L]<>0 DO
              L:=L+1;
            LFSORT((L-1),USCHD,POS);
            FOR I:= 1 TO (L-1) DO
              BEGIN
               FOR J:=(I+1) TO (L-1) DO
                BEGIN
                 IF (USCHD[I]=USCHD[J]) THEN
                    USCHD[J]:=999;
                END;
              END;
            LFSORT((L-1),USCHD,POS);

{ start local interchange procedure with low altitude supports}
            UNUM:=1;
            WHILE (USCHD[UNUM] <>0) AND (USCHD[UNUM] <>999) DO
              BEGIN
               MATCH:=FALSE;
               FLG:=0;
               USUP:=USCHD[UNUM];
              IF (USUP <>999) AND (USUP <>0) THEN
              BEGIN
               INIT_HLD(HOLD);
               NUM:=0;
               USCH_GTS(USUP,FLG,NUM,HOLD,SUPLOC,SUP);
               FIND_SUP(SUPLOC,HOLD,WGT,TURN,SUP,BVIS,EVIS,FLAG,GTSNUM);
               IF FLAG=0 THEN       {CHECK THIS OUT LATER}
                 BEGIN
                    GTS:=HOLD[GTSNUM,1];
                    NUM:=1;
                    REPEAT
                      NUM:=NUM+1;
                    UNTIL SUP[SUPLOC,NUM]=GTS;
                    SUPP:=USCHD[UNUM];
                    BEG:=BVIS[SUPLOC,NUM];
                    REQ:=EVIS[SUPLOC,NUM] - BVIS[SUPLOC,NUM];
                    TUR:=TURN[SUPLOC];
                    M:=0;
                    REPEAT
                       M:=M+1;
                    UNTIL SCHD[M,1]=0;
                   ADDSUP(SUPP,GTS,BEG,TUR,REQ,UNUM,M,SUPLOC,SCHD,FREE,USCHD);
                   LAST:=LAST+1;
                   SSORT(1,LAST,SCHD);
                 END; { FLG =0}
              USCH:=1;
```

```
WHILE (HOLD[USCH,2] <> 0) AND (MATCH=FALSE) AND (FLAG <2)
       AND (FLAG >0) DO
 BEGIN
   UGTS:=HOLD[USCH,1];
   SUPLOC:=0;
   REPEAT
    SUPLOC:=SUPLOC+1;
   UNTIL HOLD[USCH,2]=SUP[SUPLOC,1];
   IF WGT[SUPLOC] >1 THEN
    BEGIN
     SUPP:=HOLD[USCH,2];
     GTSNUM:=1;
     REPEAT
       GTSNUM:=GTSNUM+1;
       IF HOLD[USCH,1] <> SUP[SUPLOC,GTSNUM] THEN
          BEGIN
           GTS:=SUP[SUPLOC,GTSNUM];
           BEG:=BVIS[SUPLOC,GTSNUM]-TURN[SUPLOC];
           ENDD:=EVIS[SUPLOC,GTSNUM];
           CHK(BEG,ENDD,GTS,SUPP,SCHD,MATCH);
           IF MATCH=TRUE THEN
            BEGIN
              TSUP:=SUP[SUPLOC,1];
              BEG:=BVIS[SUPLOC,GTSNUM];
              REQ:=EVIS[SUPLOC,GTSNUM]-BVIS[SUPLOC,GTSNUM];
              TUR:=TURN[SUPLOC];
              UBEG:=HOLD[USCH,3];
              UTUR:=HOLD[USCH,4];
              UREQ:=HOLD[USCH,5];
              SWITCH(TSUP,GTS,BEG,REQ,TUR,USUP,UGTS,UBEG
                     ,UTUR,UREQ,SUPLOC,SCHD,USCHD);
              LAST:=LAST+1;
              SSORT(1,LAST,SCHD);
            END; {MATCH = TRUE}
         END;  {HOLD <> SUP}
     UNTIL (SUP[SUPLOC,GTSNUM]=0) OR (MATCH=TRUE);
 END;  {IF WGT[SUPLOC] >1 }
 USCH:=USCH+1;
END; {WHILE HOLD[USCH,2]<>0 AND MATCH=FALSE}
UNUM:=UNUM+1;
END;  {IF USUP = ...}
END;  {WHILE UNSCHD[UNUM]<>0}

LF_COLLECT(TDOWN,LFSCHD,LFUSCHD,LF_USCH,USCHD,POS);


(***********************************************************************)
(*                                                                     *)
(*           start insertion process now                               *)
(*                                                                     *)
```

```
(**********************************************************************)

{wipe out all unscheduled  low altitude supports                  }
{ assuming that these supports can not be scheduled and           }
{ coordination required to get them scheduled                     }
            J:=1;
            REPEAT
              USCHD[J]:=0;
              J:=J+1;
            UNTIL USCHD[J]=0;


{determine the amount of free space available at each GTS that a}
{support can be scheduled at                                    }

            FREE_SPC(SCHD,FREE);



(*data file for all high flier records records                       *)

       ASSIGN(INFILE,'C:\TP\SRSREAL\SUPHF1.DAT');
       RESET(INFILE);

       ASSIGN(INFILELK,'C:\TP\SRSREAL\LOOK.DAT');
       APPEND(INFILELK);

{ read in all high flier records                                       }

       WHILE NOT EOF(INFILE) DO
         BEGIN
           READLN(INFILE,SU,CH,TGTS,ABV,AEV,RE,TA,ID,REVV);
           NUMREC:=NUMREC+1;
           FLAG:=1;   {INDICATES THE SUPPORT IS A HIGH FLIER}
           FILLARR(SU,ABV,AEV,TA,RE,FLAG,TGTS,SGTS
                   ,SUP,BVIS,EVIS,TURN,WGT,USCHD);

           WRITELN(INFILELK,SU,CH,TGTS,CH,ID);

         END;  {WHILE NOT EOF}

         CLOSE(INFILELK);
         CLOSE(INFILE);
         J:=0;
         TOTHF:=0;
         REPEAT
           J:=J+1;
           POSN:=POS[J];
           TOTHF:=TOTHF+BVIS[POSN,1];
         UNTIL USCHD[J]=0;
         AVGHF:=TOTHF/J;
```

```
         HFILL(USCHD,POS,SUP,WGT,BVIS,EVIS);

{start insertion routine - sort hf supports by sup len, flexibility, vis}

         HFSORT(USCHD,POS,HFTOT);
         USCHD_NUM:=1;
         WHILE USCHD[USCHD_NUM]<>0 DO
           BEGIN

{**** RANK ORDER THE UNSCHEDULED SUPPORTS BY GTS WITH MOST FREE SPACE  **}
             FLAG:=1;
             POS_NUM:=POS[USCHD_NUM];
             ISERT:=FALSE;
             MIN:=2000;
             MAX:=0;
             N:=0;
             TSU:=0;

             INIT_HLD(HOLD);
             USCH_GTS(TSU,FLAG,POS_NUM,HOLD,N,SUP);
             INS_GTS(FREE,HOLD);

             MATCH:=FALSE;
             L:=0;
             REPEAT
               L:=L+1;
               GTS:=HOLD[L,1];
               INT_SPC(SPACE);
{  PUT INSERTION PROCEDURE HERE, MIN MAX REQ>MIN-5 OR MAX > REQ + 60  }
{ MATCH =TRUE                                                         }
             IF GTS <>0 THEN
              BEGIN
               FILL_SPC(GTS,SCHD,SPACE);
               N:=1;
               REPEAT
                 N:=N+1;
               UNTIL SUP[POS_NUM,N]=GTS;
               UBEG:=BVIS[POS_NUM,N]-TURN[POS_NUM];
               REQ:=BVIS[POS_NUM,1]+TURN[POS_NUM];
               ENDD:=EVIS[POS_NUM,N];
               INSERT (UBEG,ENDD,REQ,SPACE,TMAX,TMIN
                       ,TMINL,TMAXL,OK);
               IF OK=TRUE THEN
                 BEGIN
                  ISERT:=TRUE;
                  CHK_INS(REQ,GTS,TMAX,TMIN,TMAXL,TMINL,MAX,MIN
                          ,MAXL,MINL,MINGTS,MAXGTS,MATCH);
                 END;  {OK=TRUE}
              END; {GTS <>0}
             UNTIL(HOLD[L,1]=0) OR (MATCH=TRUE);
```

```
{ ureq = difference between min and req when match = false          }
{ utur = difference between max and req when match = false          }
{ OBJECT HERE IS TO MINIMIZE WASTED SPACE IN SCHEDULE                }



         IF  (MIN<>1000) AND (MAX<>0) THEN
          BEGIN
           INS_CK(MAX,MIN,REQ,MAXL,MINL,MAXGTS,MINGTS,POS_NUM,MATCH,GTS,BEG);
                   SUPP:=USCHD[USCHD_NUM];
                   REQ:=BVIS[POS_NUM,1];
                   TUR:=TURN[POS_NUM];
                   J:=0;
                   REPEAT
                      J:=J+1;
                   UNTIL SCHD[J,1]=0;
                   FLG:=1;
                    ADDSUP(SUPP,GTS,BEG,TUR,REQ,USCHD_NUM,J,POS_NUM,SCHD
                           ,FREE,USCHD);
                  SSORT(1,J,SCHD);
          END;  { MIN<>1000 AND MAX<>0}
          USCHD_NUM:=USCHD_NUM+1;
        END;    {WHILE USCHD[USCHD_NUM<>0}


(**************************************************************************)
(*                                                                        *)
(* start high flier 3 opt bumping procedure                               *)
(*                                                                        *)
(**************************************************************************)


          NUM:=1;
          WHILE USCHD[NUM]<>0 DO
            BEGIN

              IF USCHD[NUM]<>999 THEN
                BEGIN

                  INIT_HLD(HOLD);
                  FLAG:=1;
                  POS_NUM:=POS[NUM];
                  USUP:=USCHD[NUM];
                  USCH_GTS(USUP,FLAG,POS_NUM,HOLD,N,SUP);


                 OPT_INS(POS_NUM,SCHD,HOLD,SPACE,
                 USH_LOC,INT1_SUP,USH_GTS,INT1_POS,GOON,STOP);
```

```
{             says an insertion was possible}

               IF STOP=TRUE THEN
                  INT2_SUP:=999;

{ insertion not possible, but a scheduled support to interchange was found}
              IF (GOON=TRUE) AND (STOP=FALSE) THEN
               BEGIN
                 INIT_HLD(HOLD);
                 FLAG:=1;
                 USCH_GTS(INT1_SUP,FLAG,INT1_POS,HOLD,N,SUP);


                 OPT_INS(INT1_POS,SCHD,HOLD,SPACE,
                 INT1_LOC,INT2_SUP,INT1_GTS,INT2_POS,GOON,STOP);

               END;  {IF GO=TRUE AND STOP=FALSE}

{ insertion not possible, but a scheduled support to interchange was found}

                IF (GOON=TRUE) AND (STOP=FALSE) THEN
                  BEGIN
                   INIT_HLD(HOLD);
                   FLAG:=1;
                   USCH_GTS(INT2_SUP,FLAG,INT2_POS,HOLD,N,SUP);


                   OPT_INS(INT1_POS,SCHD,HOLD,SPACE,
                   OPTLOC,INT2_SUP,INT2_GTS,OPTDIFF,GOON,STOP);

                  END;  {GOON=TRUE & STOP=FALSE}

                  IF STOP=TRUE THEN

                   OPT_ADD(NUM,USUP,USH_GTS,USH_LOC,POS_NUM,INT1_SUP,
                   INT1_GTS,INT1_LOC,INT1_POS,INT2_SUP,INT2_GTS,INT2_LOC,
                   INT2_POS,SCHD,USCHD,FREE);


              END;  {IF USCHD[NUM]<>999}

              NUM:=NUM+1;
          END;  {WHILE USCHD[NUM]<>0 DO}

          J:=0;
          REPEAT
            J:=J+1;
          UNTIL SCHD[J,1]=0;
          NONFL(j,SCHD);
```

```
        J:=0;
        REPEAT
          J:=J+1;
        UNTIL SCHD[J,1]=0;
        SSORT(1,J-1,SCHD);


(**********************************************************************)
(*                                                                    *)
(*  Put in procedures that write out results of schedule              *)
(*                                                                    *)
(**********************************************************************)

      ASSIGN(INFILE,'C:\TP\SRSREAL\SCHEDULE.DAT');
      REWRITE(INFILE);

      WRT_SCHD(SCHD,INFILE);
      ASSIGN(INFILE,'C:\TP\SRSREAL\UNSCHED.DAT');
      REWRITE(INFILE);
      NUM:=0;
      HFNUM:=0;
      REPEAT
        NUM:=NUM+1;
        IF USCHD[NUM]<>999 THEN
          BEGIN
          HFNUM:=HFNUM+1;
          END;
      UNTIL USCHD[NUM]=0;
      WRT_USCHD(LF_USCH,USCHD,INFILE);
      MAX_SPC(SCHD,FRE_SPC);
      NUM:=0;
      REPEAT
        NUM:=NUM+1;
      UNTIL USCHD[NUM]=999;
      NUM:=NUM-1;
      SCH_STATS(LFSCHD,LFUSCHD,HFTOT,HFNUM,FRE_SPC,FREE,SGTS);

      NUM:=0;
      REPEAT
        NUM:=NUM+1;
      UNTIL USCHD[NUM]=0;
      NUM:=NUM-1;

END.
```

Appendix B. Sample SRS database

The data from the ASTRO general list file was stored in two databases: a low altitude database and a high altitude database. Every section of the SRS methodology (MIP, 2 interchange, insertion, and 3 interchange procedures) requires information from these databases. Each database is read in by the program and the information contained in the databases is stored in arrays Table B.1 provides sample low altitude support entries in the database. A support in this database is identified by an identification number (ident) and revolution number combination and is given a support number of identification purposes. Additionally, each entry in the database associates a specific support with a specific RTS. In Table ??, ident 2532 and revolution number 8073 would identify a support labeled 76. The label 76 is required because ident 2532 could have two support requirements during a day and the label would distinguish between each support requirement. Since support 76 is visible to two RTSs, there are two entries in the database for this support. The beginning and ending visibility times are in minutes. For example, the time 666 in the database equates to 1106 in the hhmm format.

The databases for each data set were not included in this thesis document. For each day, the two databases were thirty pages long. Including these databases would add one hundred eighty pages to the length of the document. The total length of the document would then be over three hundred and fifty pages, which is too long. However, the databases are available upon request.

Table B.1. Sample Low Altitude Database Entries

Block 1 Data							
Sup No.	RTS	BV	EV	Req Len	TAT	Ident	Rev No
76	PIKE-A	642	647	5	20	2532	8073.1
76	POGO-C	643	658	15	15	2532	8073.2
289	INDI-A	643	644	1	20	9845	9291.0
24	LION-B	645	661	16	15	1056	9756.3
188	GUAM-A	645	653	8	20	6790	2683.0
272	BOSS-B	648	663	15	20	9757	773.3
272	PIKE-A	649	664	15	20	9757	773.3
55	BOSS-A	651	668	17	20	1748	9480.1
171	POGO-A	653	668	15	15	6553	3864.2
8	POGO-A	654	671	17	20	286	2860.5
39	LION-A	654	667	13	20	1132	6856.1
8	INDI-A	657	661	4	15	4774	2583.5
55	POGO-B	660	676	16	20	1748	9480.2
289	POGO-C	663	679	16	15	9845	9291.2
95	INDI-A	667	679	12	20	3187	5786.9
171	BOSS-A	667	675	[illegible]	15	6553	3864.3
171	PIKE-A	667	682	[illegible]	20	6553	3864.3
188	POGO-C	667	683	16	15	6790	2683.2
95	GUAM-B	669	681	12	20	5821	4736.1
171	COOK-A	669	683	14	15	6553	3864.4

Appendix C. Schedule for First Data Set

Overview

This appendix contains the following information.

- Schedule of Satellite supports and RTS downtime requirements
- List of unscheduled satellite supports

Schedule for data set 1

Note: The day was divided into one minute increments. The Strrt Tme and Req Len columns are represented in minutes. For example, the start time of 1410 is equivelent to 2330. The request length of 90 is equivalent to one hour thirty minutes.

Sup	IRON	RTS	Strt Tme	Req Len	TAT
13	1056	POGO-A	51	13	20
888	DOWN	POGO-A	67	70	0
14	1056	POGO-A	150	13	20
391	6374	POGO-A	199	15	15
98	6553	POGO-A	234	14	20
72	4774	POGO-A	268	11	20
409	7225	POGO-A	348	175	15
139	9757	POGO-A	559	16	19
417	7304	POGO-A	591	10	15
104	6553	POGO-A	625	14	20
77	4774	POGO-A	660	14	20
30	1056	POGO-A	756	16	20
888	DOWN	POGO-A	772	67	0
81	4774	POGO-A	854	14	15
380	6012	POGO-A	915	13	15
109	6553	POGO-A	1024	12	15
148	9757	POGO-A	1062	12	15
421	7304	POGO-A	1090	15	15
111	6553	POGO-A	1123	13	15
152	9757	POGO-A	1161	13	20
888	DOWN	POGO-A	1164	41	0
113	6553	POGO-A	1221	14	15
154	9757	POGO-A	1260	14	20
157	9757	POGO-A	1359	16	20

378	6012	POGO-B	15	20	15
888	DOWN	POGO-B	105	130	0
17	1056	POGO-B	249	15	20
100	6553	POGO-B	331	14	15
73	4774	POGO-B	367	12	15
888	DOWN	POGO-B	379	50	0
20	1056	POGO-B	449	17	20
452	8896	POGO-B	482	40	15
24	1056	POGO-B	551	16	15
448	8639	POGO-B	583	48	15
26	1056	POGO-B	653	16	20
416	7304	POGO-B	685	47	15
79	4774	POGO-B	757	14	20
418	7304	POGO-B	787	10	15
888	DOWN	POGO-B	800	180	0
494	7506	POGO-B	1020	15	15
419	7304	POGO-B	1051	15	15
413	7225	POGO-B	1085	10	15
313	2524	POGO-B	1111	45	15
440	7506	POGO-B	1172	10	15
497	470	POGO-B	1225	15	15
888	DOWN	POGO-B	1243	60	0
114	6553	POGO-B	1318	15	15
40	1056	POGO-B	1362	13	15
49	2532	POGO-C	26	16	15
888	DOWN	POGO-C	41	110	0
71	4774	POGO-C	169	10	15
158	9845	POGO-C	217	16	15
60	3187	POGO-C	326	16	15
4	286	POGO-C	364	12	15
463	9434	POGO-C	392	15	15
61	3187	POGO-C	426	17	15
5	286	POGO-C	466	14	15
427	7310	POGO-C	496	10	15
62	3187	POGO-C	527	16	15
6	286	POGO-C	567	16	15
319	3055	POGO-C	599	19	15
53	2532	POGO-C	638	13	15
7	286	POGO-C	669	16	15
428	7310	POGO-C	701	20	15
5[illegible]	2532	POGO-C	739	14	15
888	DOWN	POGO-C	753	66	0
64	3187	POGO-C	834	14	15
65	3187	POGO-C	939	11	15
8	286	POGO-C	971	16	15
430	7310	POGO-C	1003	20	15
86	4774	POGO-C	1047	14	20
9	286	POGO-C	1071	16	10
382	6012	POGO-C	1103	25	15
67	3187	POGO-C	1145	13	15

10	286	POGO-C	1171	17	10
68	3187	POGO-C	1247	15	15
57	2532	POGO-C	1335	16	15
888	DOWN	POGO-C	1350	90	0
486	6012	HULA-A	15	60	15
443	7837	HULA-A	91	35	15
888	DOWN	HULA-A	200	115	0
135	9757	HULA-A	335	16	20
487	6012	HULA-A	367	150	15
888	DOWN	HULA-A	520	163	0
406	6738	HULA-A	699	15	15
29	1056	HULA-A	736	16	20
438	7506	HULA-A	768	15	15
106	6553	HULA-A	807	13	20
491	9445	HULA-A	836	25	15
84	4774	HULA-A	970	11	20
467	9443	HULA-A	1000	10	15
150	9757	HULA-A	1079	16	20
888	DOWN	HULA-A	1085	62	0
326	3310	HULA-A	1163	45	15
425	7304	HULA-A	1224	10	15
499	6142	HULA-A	1324	25	15
454	9363	HULA-A	1370	10	20
383	6012	HULA-A	1401	35	15
384	6071	HULA-B	15	30	15
2	286	HULA-B	139	14	20
400	6451	HULA-B	200	15	15
3	286	HULA-B	238	16	20
464	9441	HULA-B	270	15	15
403	6453	HULA-B	300	15	15
134	9757	HULA-B	335	16	20
377	5953	HULA-B	367	20	15
51	2532	HULA-B	413	17	20
888	DOWN	HULA-B	430	65	0
52	2532	HULA-B	515	12	20
119	6790	HULA-B	551	10	20
434	7314	HULA-B	577	30	15
432	7314	HULA-B	623	15	15
28	1056	HULA-B	736	16	20
305	1864	HULA-B	773	10	20
105	6553	HULA-B	807	13	20
888	DOWN	HULA-B	820	64	0
107	6553	HULA-B	904	13	20
341	4373	HULA-B	933	15	15
83	4774	HULA-B	970	11	20
66	3187	HULA-B	1020	16	20
149	9757	HULA-B	1079	16	20
490	9445	HULA-B	1111	35	15
47	1132	HULA-B	1169	12	20

120	6790	HULA-B	1222	12	20
56	2532	HULA-B	1257	11	20
888	DOWN	HULA-B	1268	130	0
375	5953	HULA-B	1414	20	15
445	8275	COOK-A	15	15	15
70	4774	COOK-A	54	13	20
888	DOWN	COOK-A	80	60	0
387	6142	COOK-A	155	87	15
314	2524	COOK-A	283	15	15
317	3028	COOK-A	314	45	15
385	6071	COOK-A	375	10	15
397	6392	COOK-A	400	10	15
123	7050	COOK-A	441	11	20
888	DOWN	COOK-A	452	69	0
22	1056	COOK-A	539	15	20
410	7225	COOK-A	570	190	15
411	7225	COOK-A	776	15	15
888	DOWN	COOK-A	871	220	0
379	6012	COOK-A	1107	35	15
492	7310	COOK-A	1158	10	15
128	7050	COOK-A	1192	14	20
129	7050	COOK-A	1293	15	20
442	7641	COOK-A	1324	20	15
501	8639	COOK-A	1371	15	15
91	4774	COOK-A	1425	15	15
482	9794	COOK-B	15	15	15
95	6553	COOK-B	54	14	15
386	6071	COOK-B	84	57	15
312	2272	COOK-B	200	15	15
447	8275	COOK-B	231	30	15
485	8639	COOK-B	277	15	15
43	1132	COOK-B	349	15	20
888	DOWN	COOK-B	354	242	0
412	7225	COOK-B	612	45	15
401	6451	COOK-B	673	30	15
93	6142	COOK-B	792	16	15
398	6394	COOK-B	824	30	15
144	9757	COOK-B	874	16	20
146	9757	COOK-B	975	14	20
888	DOWN	COOK-B	989	76	0
414	7225	COOK-B	1081	175	15
38	1056	COOK-B	1274	16	15
888	DOWN	COOK-B	1340	75	0
117	6553	COOK-B	1425	15	15
94	6553	INDI-A	13	15	15
449	8701	INDI-A	44	49	15
362	5329	INDI-A	109	35	15
133	9757	INDI-A	179	16	20

888	DOWN	INDI-A	195	105	0
323	3160	INDI-A	316	20	15
324	3160	INDI-A	352	20	15
371	5775	INDI-A	400	20	15
450	8896	INDI-A	436	40	15
372	5775	INDI-A	500	20	15
888	DOWN	INDI-A	521	184	0
458	9365	INDI-A	721	58	15
420	7304	INDI-A	795	145	15
888	DOWN	INDI-A	940	71	0
46	1132	INDI-A	1025	14	20
346	4832	INDI-A	1127	10	15
374	5775	INDI-A	1170	100	15
368	5775	INDI-A	1286	20	15
39	1056	INDI-A	1333	16	20
116	6553	INDI-A	1389	12	15
344	4524	INDI-A	1417	10	15
308	2124	BOSS-A	15	20	15
338	4035	BOSS-A	51	10	15
461	9366	BOSS-A	77	30	15
309	2124	BOSS-A	123	20	15
888	DOWN	BOSS-A	200	201	0
102	6553	BOSS-A	421	13	20
74	4774	BOSS-A	475	12	20
475	9794	BOSS-A	503	15	15
316	2941	BOSS-A	537	15	15
75	4774	BOSS-A	571	15	15
408	7225	BOSS-A	602	10	15
141	9757	BOSS-A	670	16	20
888	DOWN	BOSS-A	686	63	0
45	1132	BOSS-A	769	14	20
423	7304	BOSS-A	799	115	15
33	1056	BOSS-A	970	13	20
330	3726	BOSS-A	1004	45	20
307	2124	BOSS-A	1065	20	15
393	6391	BOSS-A	1100	10	15
415	7225	BOSS-A	1141	235	15
303	712	BOSS-A	1397	30	20
130	9757	BOSS-B	39	16	19
888	DOWN	BOSS-B	54	140	0
92	5821	BOSS-B	163	12	20
339	4035	BOSS-B	210	30	15
335	4035	BOSS-B	256	10	15
394	6391	BOSS-B	282	30	15
18	1056	BOSS-B	340	15	20
345	4524	BOSS-B	371	30	15
304	712	BOSS-B	422	15	20
315	2567	BOSS-B	477	45	15
433	7314	BOSS-B	538	30	15

340	4035	BOSS-B	600	10	15
44	1132	BOSS-B	668	15	20
381	6012	BOSS-B	701	50	15
142	9757	BOSS-B	771	15	20
888	DOWN	BOSS-B	786	60	0
55	2532	BOSS-B	848	15	15
390	6280	BOSS-B	882	30	15
441	7506	BOSS-B	1028	20	15
35	1056	BOSS-B	1070	16	20
888	DOWN	BOSS-B	1100	111	0
89	4774	BOSS-B	1232	15	15
11	286	BOSS-B	1264	16	15
389	6280	BOSS-B	1296	10	15
115	6553	BOSS-B	1327	13	20
12	286	BOSS-B	1366	14	5
503	1920	BOSS-B	1396	15	15
58	2532	BOSS-B	1426	14	13
888	DOWN	LION-A	0	120	0
364	5681	LION-A	136	45	15
360	5329	LION-A	197	20	15
363	5329	LION-A	233	55	15
460	9366	LION-A	304	30	15
342	4373	LION-A	350	45	15
301	470	LION-A	412	45	15
396	6392	LION-A	473	30	15
888	DOWN	LION-A	518	182	0
369	5775	LION-A	716	20	15
459	9366	LION-A	752	30	15
124	7050	LION-A	806	15	20
80	4774	LION-A	845	12	15
108	6553	LION-A	933	14	15
126	7050	LION-A	1010	15	20
147	9757	LION-A	1052	11	20
151	9757	LION-A	1150	16	12
373	5775	LION-A	1182	20	15
347	4832	LION-A	1218	10	15
153	9757	LION-A	1252	15	15
302	712	LION-A	1288	10	20
343	4524	LION-A	1330	15	15
156	9757	LION-A	1358	9	12
465	9442	LION-A	1383	10	15
361	5329	LION-B	15	20	15
888	DOWN	LION-B	100	105	0
97	6553	LION-B	225	14	15
42	1132	LION-B	271	13	14
334	4035	LION-B	300	10	15
99	6553	LION-B	325	11	15
136	9757	LION-B	365	16	15
453	8896	LION-B	397	45	15

137	9757	LION-B	466	16	20
888	DOWN	LION-B	482	117	0
336	4035	LION-B	615	25	15
27	1056	LION-B	665	13	15
311	2124	LION-B	700	20	15
337	4035	LION-B	800	10	15
31	1056	LION-B	867	15	20
125	7050	LION-B	907	15	20
82	4774	LION-B	940	15	15
422	7304	LION-B	971	170	15
349	4845	LION-B	1157	30	15
348	4845	LION-B	1203	30	15
888	DOWN	LION-B	1259	181	0
888	DOWN	GUAM-A	0	88	0
121	7050	GUAM-A	108	15	20
365	5681	GUAM-A	139	15	15
15	1056	GUAM-A	173	14	15
320	3160	GUAM-A	203	20	15
484	3028	GUAM-A	239	15	15
480	7304	GUAM-A	270	15	15
404	6453	GUAM-A	300	30	15
101	6553	GUAM-A	355	14	15
888	DOWN	GUAM-A	369	123	0
436	7506	GUAM-A	508	10	15
138	9757	GUAM-A	535	15	15
468	9444	GUAM-A	566	35	15
140	9757	GUAM-A	636	15	15
325	3310	GUAM-A	667	15	15
370	5775	GUAM-A	700	15	15
888	DOWN	GUAM-A	800	120	0
32	1056	GUAM-A	935	16	15
466	9443	GUAM-A	1000	15	15
34	1056	GUAM-A	1039	11	15
888	DOWN	GUAM-A	1070	130	0
496	7225	GUAM-A	1216	125	15
367	5775	GUAM-A	1357	20	15
495	7225	GUAM-A	1393	15	15
888	DOWN	GUAM-B	0	200	0
321	3160	GUAM-B	216	20	15
322	3160	GUAM-B	25[illegible]	20	15
405	6453	GUAM-B	300	30	15
888	DOWN	GUAM-B	351	123	0
328	3726	GUAM-B	485	195	10
399	6394	GUAM-B	696	10	15
888	DOWN	GUAM-B	714	104	0
124	7050	GUAM-B	838	13	20
470	9445	GUAM-B	867	25	15
376	5953	GUAM-B	980	20	15
110	6553	GUAM-B	1098	14	20

888 DOWN	GUAM-B	1111	149	0
88 4774	GUAM-B	1168	13	13
155 9757	GUAM-B	1283	16	20
48 1132	GUAM-B	1366	14	20
446 8275	PIKE-A	15	30	15
310 2124	PIKE-A	61	20	15
132 9757	PIKE-A	138	16	12
888 DOWN	PIKE-A	154	64	0
122 7050	PIKE-A	238	15	20
462 9366	PIKE-A	269	15	15
402 6451	PIKE-A	300	30	15
483 1920	PIKE-A	346	45	15
407 7225	PIKE-A	407	10	15
19 1056	PIKE-A	440	14	20
456 9364	PIKE-A	475	15	20
23 1056	PIKE-A	539	17	19
429 7310	PIKE-A	581	10	15
103 6553	PIKE-A	614	15	15
888 DOWN	PIKE-A	702	137	0
444 7837	PIKE-A	855	35	15
332 3726	PIKE-A	960	165	10
431 7310	PIKE-A	1141	10	15
36 1056	PIKE-A	1172	16	20
888 DOWN	PIKE-A	1198	67	0
455 9364	PIKE-A	1300	30	20
424 7304	PIKE-A	1346	25	15
502 8639	PIKE-A	1389	45	15
888 DOWN	REEF-A	0	61	0
131 9757	REEF-A	81	12	20
435 7506	REEF-A	109	35	15
41 1132	REEF-A	193	14	20
474 9783	REEF-A	223	15	15
473 9521	REEF-A	253	73	15
469 9444	REEF-A	342	25	15
327 3726	REEF-A	378	45	[illegible]
21 1056	REEF-A	480	16	15
306 1920	REEF-A	512	10	15
25 1056	REEF-A	582	14	20
76 4774	REEF-A	632	12	15
888 DOWN	REEF-A	644	64	0
78 4774	REEF-A	728	14	15
451 8896	REEF-A	758	40	15
143 9757	REEF-A	831	16	20
488 5329	REEF-A	863	20	15
472 9446	REEF-A	899	15	15
145 9757	REEF-A	933	13	15
888 DOWN	REEF-A	955	145	0
439 7506	REEF-A	1116	10	15
489 5329	REEF-A	1142	20	15

471	9446	REEF-A	1178	30	15
37	1056	REEF-A	1231	17	20
366	5775	REEF-A	1264	20	15
493	2567	REEF-A	1300	15	15
318	3026	REEF-A	1331	15	15
90	4774	REEF-A	1370	14	20
888	DOWN	REEF-A	1394	56	0

Unscheduled Supports

The following supports could not be scheduled.

Low Altitude Satellite Supports

2532

3187(2)

4774(2)

6573

7050

Medium/High Altitude Satellite Supports

5037

5037

3726

3726

6391

3055

5037

6280

7301

4845

4845

4955

4955
5037
5037
5037
3310
6391
3310
9364
7225
7506
2524
7310

Appendix D. Schedule for Second Data Set

Overview

This appendix contains the following information.

- Schedule of Satellite supports and RTS downtime requirements
- List of unscheduled satellite supports

Schedule for data set 2

Note: The day was divided into one minute increments. The Strt Tme and Req Len columns are represented in minutes. For example, the start time of 1410 is equivelent to 2330. The request length of 90 is equivalent to one hour thirty minutes.

Sup	IRON	RTS	Srt Tme	Req Len	Turn Tme
307	1920	POGO-A	21	15	15
480	7506	POGO-A	52	15	15
888	DOWN	POGO-A	190	115	0
131	9757	POGO-A	325	16	20
412	7225	POGO-A	357	7	15
413	7225	POGO-A	443	15	15
20	1056	POGO-A	521	16	15
305	470	POGO-A	553	45	15
22	1056	POGO-A	623	16	20
448	8639	POGO-A	655	48	15
379	6012	POGO-A	710	13	15
88	4774	POGO-A	787	14	18
26	1056	POGO-A	827	16	20
90	4774	POGO-A	834	14	15
29	1056	POGO-A	929	16	15
888	DOWN	POGO-A	945	168	0
147	9757	POGO-A	1133	12	15
381	6012	POGO-A	1161	25	15
444	7837	POGO-A	1206	35	15
473	9783	POGO-A	1295	15	15
888	DOWN	POGO-A	1310	90	0
54	1748	POGO-A	1420	15	20

41	1132	POGO-B	3	6	20
888	DOWN	POGO-B	29	223	0
326	3310	POGO-B	268	45	15
407	6738	POGO-B	329	15	15
17	1056	POGO-B	419	16	15
102	6553	POGO-B	460	15	15
450	8896	POGO-B	491	40	15
52	1748	POGO-B	551	6	20
888	DOWN	POGO-B	556	115	0
86	4774	POGO-B	690	15	20
422	7304	POGO-B	721	205	15
419	7304	POGO-B	942	15	15
144	9757	POGO-B	1034	12	15
33	1056	POGO-B	1132	15	20
888	DOWN	POGO-B	1147	70	0
36	1056	POGO-B	1232	14	20
151	9757	POGO-B	1331	15	20
460	9434	POGO-B	1362	30	15
888	DOWN	POGO-C	[illegible]	100	0
363	5681	POGO-C	126	15	15
507	1920	POGO-C	157	15	15
426	7310	POGO-C	188	20	15
116	6790	POGO-C	308	16	20
888	DOWN	POGO-C	322	116	0
3	286	POGO-C	453	14	15
410	7225	POGO-C	483	10	15
69	3187	POGO-C	512	17	15
4	286	POGO-C	554	16	15
59	2532	POGO-C	607	12	15
5	286	POGO-C	656	16	15
60	2532	POGO-C	707	14	15
888	DOWN	POGO-C	721	66	0
61	2532	POGO-C	808	15	15
109	6553	POGO-C	858	10	15
62	2532	POGO-C	907	16	15
6	286	POGO-C	958	16	20
888	DOWN	POGO-C	973	27	0
117	6790	POGO-C	1015	12	15
7	286	POGO-C	1058	17	15
73	3187	POGO-C	1131	12	15
8	286	POGO-C	1159	16	15
64	2532	POGO-C	1205	16	15
74	3187	POGO-C	1233	14	12
9	286	POGO-C	1260	16	5
65	2532	POGO-C	1304	16	4
75	3187	POGO-C	1334	16	5
10	286	POGO-C	1362	16	12
66	2532	POGO-C	1404	16	14
475	6394	HULA-A	15	5	15

12 1056	HULA-A	40	14	20
888 DOWN	HULA-A	54	151	0
2 286	HULA-A	225	16	20
484 8639	HULA-A	257	15	15
486 6142	HULA-A	268	15	15
433 7314	HULA-A	320	30	15
406 6453	HULA-A	366	15	15
133 9757	HULA-A	408	15	18
313 2272	HULA-A	439	45	15
464 9443	HULA-A	500	10	15
403 6453	HULA-A	530	30	15
408 6738	HULA-A	576	45	15
24 1056	HULA-A	706	16	20
888 DOWN	HULA-A	733	161	0
374 5953	HULA-A	920	20	15
451 8896	HULA-A	957	30	15
434 7314	HULA-A	1003	15	15
145 9757	HULA-A	1050	17	20
46 1132	HULA-A	1101	15	20
423 7304	HULA-A	1217	10	15
376 5953	HULA-A	1243	20	15
424 7304	HULA-A	1279	15	15
888 DOWN	HULA-A	1376	64	0
469 9445	HULA-B	15	10	15
314 2272	HULA-B	120	15	15
472 9521	HULA-B	151	15	15
397 6394	HULA-B	250	30	15
130 9757	HULA-B	307	15	15
888 DOWN	HULA-B	320	43	0
57 2532	HULA-B	383	15	20
465 9443	HULA-B	430	30	15
58 2532	HULA-B	482	15	20
121 7050	HULA-B	570	13	20
443 7837	HULA-B	599	35	15
888 DOWN	HULA-B	640	90	0
107 6553	HULA-B	747	3	15
71 3187	HULA-B	907	13	20
76 3726	HULA-B	947	10	20
72 3187	HULA-B	1005	17	20
405 6453	HULA-B	1060	30	15
401 6451	HULA-B	1106	30	15
506 1864	HULA-B	1160	10	20
888 DOWN	HULA-B	1171	80	0
390 6374	HULA-B	1267	15	15
404 6453	HULA-B	1340	10	15
398 6394	HULA-B	1366	10	15
383 6012	HULA-B	1397	35	15
888 DOWN	COOK-A	71	118	0
129 9757	COOK-A	209	17	20

483 7304	COOK-A	242	15	15
42 1132	COOK-A	282	15	20
51 1748	COOK-A	334	17	20
446 8275	COOK-A	367	10	15
19 1056	COOK-A	509	14	20
394 6392	COOK-A	539	190	15
428 7310	COOK-A	745	10	15
437 7506	COOK-A	771	15	15
888 DOWN	COOK-A	858	68	0
142 9757	COOK-A	946	16	20
330 3726	COOK-A	973	240	10
888 DOWN	COOK-A	1220	220	0
384 6071	COOK-B	15	30	15
455 9364	COOK-B	61	15	15
319 3028	COOK-B	154	45	15
50 1748	COOK-B	239	15	20
364 5681	COOK-B	274	30	15
402 6451	COOK-B	330	30	15
386 6142	COOK-B	376	15	15
409 7225	COOK-B	411	10	15
442 7641	COOK-B	500	42	15
888 DOWN	COOK-B	626	175	0
317 2567	COOK-B	817	15	15
396 6392	COOK-B	850	30	15
429 7310	COOK-B	896	20	15
45 1132	COOK-B	1002	10	20
329 3726	COOK-B	1040	40	20
456 9364	COOK-B	1110	15	15
373 5953	COOK-B	1140	20	15
440 7506	COOK-B	1176	10	15
452 8896	COOK-B	1202	25	15
37 1056	COOK-B	1244	16	15
498 9794	COOK-B	1276	15	15
496 7304	COOK-B	1307	15	15
39 1056	COOK-B	1346	13	15
115 6553	COOK-B	1364	11	5
888 DOWN	COOK-B	1375	65	0
77 4774	INDI-A	27	14	15
347 4845	INDI-A	54	30	12
340 4035	INDI-A	97	30	12
888 DOWN	INDI-A	154	162	0
449 8701	INDI-A	332	49	15
97 5821	INDI-A	417	12	20
470 9446	INDI-A	440	30	10
418 7304	INDI-A	497	10	15
471 9446	INDI-A	520	10	10
21 1056	INDI-A	550	16	20
416 7304	INDI-A	582	10	15
457 9365	INDI-A	610	57	15

106 6553	INDI-A	687	13	20
888 DOWN	INDI-A	700	60	0
108 6553	INDI-A	784	12	20
155 9845	INDI-A	825	16	20
335 4035	INDI-A	854	10	12
387 6280	INDI-A	869	10	4
393 6391	INDI-A	950	30	10
143 9757	INDI-A	1007	12	15
306 1864	INDI-A	1040	10	20
63 2532	INDI-A	1078	15	20
392 6391	INDI-A	1110	30	10
493 3726	INDI-A	1156	50	15
339 4035	INDI-A	1219	10	12
494 7506	INDI-A	1245	15	15
38 1056	INDI-A	1302	17	20
338 4035	INDI-A	1332	30	12
95 4774	INDI-A	1402	11	20
439 7506	INDI-A	1429	10	15
309 2124	BOSS-A	10	20	10
302 712	BOSS-A	46	15	15
391 6391	BOSS-A	72	10	10
488 3310	BOSS-A	98	15	15
888 DOWN	BOSS-A	120	100	0
395 6392	BOSS-A	236	80	15
462 9441	BOSS-A	340	25	15
436 7506	BOSS-A	381	10	15
344 4524	BOSS-A	410	30	15
458 9366	BOSS-A	460	15	15
82 4774	BOSS-A	505	13	20
83 4774	BOSS-A	602	14	15
138 9757	BOSS-A	641	15	20
888 DOWN	BOSS-A	655	67	0
139 9757	BOSS-A	742	16	20
353 5037	BOSS-A	774	300	15
399 6451	BOSS-A	1090	15	15
342 4524	BOSS-A	1120	30	15
453 8896	BOSS-A	1166	40	15
888 DOWN	BOSS-A	1210	93	0
430 7310	BOSS-A	1355	10	15
888 DOWN	BOSS-A	1367	73	0
126 9757	BOSS-B	10	16	15
96 5821	BOSS-B	87	13	20
49 1748	BOSS-B	141	17	20
56 2532	BOSS-B	189	10	14
311 2124	BOSS-B	210	20	10
343 4524	BOSS-B	250	15	15
15 1056	BOSS-B	310	14	20
888 DOWN	BOSS-B	324	66	0
16 1056	BOSS-B	410	16	20

101	6553	BOSS-B	452	14	20
68	3187	BOSS-B	507	12	15
318	2941	BOSS-B	535	15	15
417	7304	BOSS-B	572	15	15
70	3187	BOSS-B	604	17	15
389	6280	BOSS-B	626	30	4
447	8275	BOSS-B	680	30	15
421	7304	BOSS-B	726	195	15
30	1056	BOSS-B	940	11	15
356	5037	BOSS-B	1000	20	15
32	1056	BOSS-B	1040	16	20
888	DOWN	BOSS-B	1056	55	0
400	6451	BOSS-B	1127	10	15
113	6553	BOSS-B	1165	12	20
454	9363	BOSS-B	1200	10	15
357	5037	BOSS-B	1350	10	15
385	6071	BOSS-B	1376	10	15
153	9757	BOSS-B	1422	16	20
11	1056	LION-A	11	13	15
888	DOWN	LION-A	24	166	0
336	4035	LION-A	203	10	12
351	4955	LION-A	229	51	15
358	5037	LION-A	350	10	15
341	4373	LION-A	376	15	15
388	6280	LION-A	400	74	4
320	3055	LION-A	490	30	15
310	2124	LION-A	531	20	10
316	2567	LION-A	573	35	15
459	9366	LION-A	624	15	15
334	4035	LION-A	652	10	12
349	4845	LION-A	700	30	12
122	7050	LION-A	737	12	7
378	6012	LION-A	765	35	15
124	7050	LION-A	836	15	20
110	6553	LION-A	868	12	15
888	DOWN	LION-A	880	60	0
111	6553	LION-A	965	15	15
463	9442	LION-A	996	63	15
414	7225	LION-A	1158	10	15
148	9757	LION-A	1223	16	13
888	DOWN	LION-A	1239	55	0
53	1748	LION-A	1314	16	20
47	1132	LION-A	1369	15	20
40	1056	LION-A	1422	11	17
888	DOWN	LION-B	0	120	0
354	5037	LION-B	136	10	15
355	5037	LION-B	162	10	15
78	4774	LION-B	209	13	20
100	6553	LION-B	257	14	20

79	4774	LION-B	306	14	15
132	9757	LION-B	336	15	15
348	4845	LION-B	370	10	12
81	4774	LION-B	404	10	14
134	9757	LION-B	437	16	15
888	DOWN	LION-B	453	67	0
136	9757	LION-B	540	11	15
504	470	LION-B	567	15	15
333	4035	LION-B	600	10	12
23	1056	LION-B	635	11	15
420	7304	LION-B	662	10	15
98	5821	LION-B	693	13	20
25	1056	LION-B	735	16	15
361	5329	LION-B	762	20	10
312	2124	LION-B	800	20	10
27	1056	LION-B	837	15	15
89	4774	LION-B	875	13	15
888	DOWN	LION-B	888	63	0
91	4774	LION-B	971	14	18
346	4832	LION-B	995	10	9
345	4832	LION-B	1015	10	9
112	6553	LION-B	1063	12	20
146	9757	LION-B	1122	15	20
415	7225	LION-B	1158	250	15
445	8275	LION-B	1424	15	15
55	2532	GUAM-A	20	13	20
365	5775	GUAM-A	49	20	15
366	5775	GUAM-A	85	20	15
13	1056	GUAM-A	143	13	20
888	DOWN	GUAM-A	156	67	0
14	1056	GUAM-A	243	16	20
370	5775	GUAM-A	275	20	15
479	3726	GUAM-A	311	45	15
80	4774	GUAM-A	373	14	15
482	7225	GUAM-A	403	70	15
135	9757	GUAM-A	507	14	20
481	7225	GUAM-A	537	15	15
137	9757	GUAM-A	607	16	15
43	1132	GUAM-A	669	15	20
505	470	GUAM-A	700	45	15
123	7050	GUAM-A	766	15	20
888	DOWN	GUAM-A	802	93	0
28	1056	GUAM-A	905	16	20
492	7310	GUAM-A	[illegible]	10	15
332	3726	GUAM-A	960	32	10
31	1056	GUAM-A	1007	14	15
331	3726	GUAM-A	1032	165	10
369	5775	GUAM-A	1220	20	15
304	470	GUAM-A	1256	15	15
118	6790	GUAM-A	1300	13	20

888	DOWN	GUAM-A	1322	118	0
375	5953	GUAM-B	15	20	15
477	3726	GUAM-B	51	240	15
888	DOWN	GUAM-B	316	91	0
328	3726	GUAM-B	418	250	10
467	9444	GUAM-B	684	25	15
327	3726	GUAM-B	720	190	10
888	DOWN	GUAM-B	920	160	0
495	7225	GUAM-B	1096	15	15
431	7310	GUAM-B	1136	10	15
149	9757	GUAM-B	1254	16	20
888	DOWN	GUAM-B	1270	67	0
152	9757	GUAM-B	1357	13	20
500	3028	GUAM-B	1386	15	15
497	2524	GUAM-B	1417	15	15
461	9434	PIKE-A	15	10	15
888	DOWN	PIKE-A	30	60	0
127	9757	PIKE-A	110	16	20
120	7050	PIKE-A	167	15	20
411	7225	PIKE-A	198	250	15
103	6553	PIKE-A	550	12	20
84	4774	PIKE-A	605	12	15
105	6553	PIKE-A	646	14	15
888	DOWN	PIKE-A	659	29	0
87	4774	PIKE-A	701	15	15
308	2124	PIKE-A	727	20	10
141	9757	PIKE-A	844	16	15
325	3310	PIKE-A	876	15	15
303	712	PIKE-A	907	30	15
888	DOWN	PIKE-A	1000	111	0
34	1056	PIKE-A	1142	15	12
301	712	PIKE-A	1173	30	15
125	7050	PIKE-A	1225	15	20
93	4774	PIKE-A	1266	10	20
94	4774	PIKE-A	1360	15	15
499	1920	PIKE-A	1391	45	15
322	3160	REEF-A	15	20	15
323	3160	REEF-A	51	20	15
367	5775	REEF-A	87	20	15
128	9757	REEF-A	151	16	20
435	7506	REEF-A	183	35	15
468	9445	REEF-A	260	30	15
324	3160	REEF-A	306	20	15
487	3055	REEF-A	342	15	15
427	7310	REEF-A	423	10	15
18	1056	REEF-A	451	14	15
888	DOWN	REEF-A	469	88	0
104	6553	REEF-A	591	11	16

85 4774	REEF-A	661	15	15
360 5329	REEF-A	687	20	10
350 4645	REEF-A	720	30	12
359 5329	REEF-A	761	20	10
140 9757	REEF-A	803	15	20
44 1132	REEF-A	859	13	19
362 5329	REEF-A	883	90	10
888 DOWN	REEF-A	975	67	0
368 5775	REEF-A	1058	20	15
315 2524	REEF-A	1094	15	15
35 1056	REEF-A	1201	16	20
474 9794	REEF-A	1233	20	15
114 6553	REEF-A	1322	12	15
888 DOWN	REEF-A	1396	44	0

Unscheduled Supports

The following supports could not be scheduled.

Low Altitude Satellite Supports

5821

3187

9757(3)

6790

Medium/High Altitude Satellite Supports

7314

6012

6012

7304

5775

7506

6012

5037

3726

4035
9144
6012
8639
3055
6012
6012
2124
2124
2124
8639
3726
5775

Appendix E. Schedule for Third Data Set

Overview

This appendix contains the following information.

- Schedule of Satellite supports and RTS downtime requirements
- List of unscheduled satellite supports

Schedule for data set 3

Note: The day was divided into one minute segments. The Strt Tme and Req Len columns are represented in minutes. For example, the start time of 1410 is equivelent to 2330. The request length of 90 is equivalent of one hour thirty minutes.

Sup	IRON	RTS	Srt Tme	Req Len	Turn Tme
888	DOWN	POGO-A	0	88	0
101	6553	POGO-A	102	14	15
79	4774	POGO-A	225	11	12
307	1920	POGO-A	263	10	15
313	2272	POGO-A	290	15	15
81	4774	POGO-A	324	12	15
317	2941	POGO-A	352	15	15
20	1056	POGO-A	399	16	20
437	7310	POGO-A	426	20	10
493	9794	POGO-A	462	15	15
23	1056	POGO-A	500	16	20
888	DOWN	POGO-A	516	52	0
107	6553	POGO-A	588	14	15
426	7304	POGO-A	618	15	15
142	9757	POGO-A	719	16	20
90	4774	POGO-A	812	14	20
91	4774	POGO-A	908	14	15
420	7225	POGO-A	938	15	15
478	9434	POGO-A	990	15	15
115	6553	POGO-A	1086	13	20
152	9757	POGO-A	1119	13	15
888	DOWN	POGO-A	1143	59	0
46	1748	POGO-A	1222	12	17
513	470	POGO-A	1250	15	15

116	6553	POGO-A	1232	14	15
157	9757	POGO-A	1317	16	15
302	712	POGO-A	1354	15	20
48	1748	POGO-A	1413	14	20
130	9757	POGO-B	107	17	20
888	DOWN	POGO-B	124	71	0
133	9757	POGO-B	210	16	15
18	1056	POGO-B	298	16	15
98	5821	POGO-B	382	9	20
83	4774	POGO-B	422	14	20
468	8896	POGO-B	452	40	15
138	9757	POGO-B	517	16	20
430	7304	POGO-B	549	15	15
464	8639	POGO-B	580	15	15
86	4774	POGO-B	617	15	20
54	2532	POGO-B	682	14	15
490	9521	POGO-B	712	90	15
111	6553	POGO-B	888	11	20
122	6790	POGO-B	924	14	15
113	6553	POGO-B	988	11	15
149	9757	POGO-B	1020	13	15
888	DOWN	POGO-B	1038	46	0
93	4774	POGO-B	1101	15	15
45	1748	POGO-B	1126	11	10
154	9757	POGO-B	1218	14	20
34	1056	POGO-B	1312	13	16
99	5821	POGO-B	1353	14	20
888	DOWN	POGO-B	1370	70	0
49	2532	POGO-C	70	16	5
62	3187	POGO-C	153	17	15
888	DOWN	POGO-C	170	89	0
63	3187	POGO-C	254	16	15
4	286	POGO-C	298	12	15
495	7310	POGO-C	326	10	15
64	3187	POGO-C	354	16	15
5	286	POGO-C	401	13	15
66	3187	POGO-C	454	17	15
84	4774	POGO-C	520	14	15
67	3187	POGO-C	555	17	15
53	2532	POGO-C	581	12	5
120	6790	POGO-C	620	16	15
467	8896	POGO-C	652	25	15
55	2532	POGO-C	782	15	5
121	6790	POGO-C	822	15	15
69	3187	POGO-C	863	13	15
56	2532	POGO-C	882	16	5
888	DOWN	POGO-C	898	90	0
9	286	POGO-C	1007	16	15
10	286	POGO-C	1107	17	15

73	3187	POGO-C	1174	13	15
11	286	POGO-C	1208	17	15
123	6790	POGO-C	1234	12	9
59	2532	POGO-C	1279	16	15
12	286	POGO-C	1310	16	15
888	DOWN	POGO-C	1326	38	0
60	2532	POGO-C	1379	16	15
13	286	POGO-C	1413	15	8
318	3028	HULA-A	34	45	15
486	9445	HULA-A	95	30	15
450	7506	HULA-A	248	10	15
3	286	HULA-A	275	15	15
505	6142	HULA-A	306	15	15
39	1132	HULA-A	408	15	20
888	DOWN	HULA-A	423	84	0
508	6012	HULA-A	523	15	15
407	6394	HULA-A	560	30	15
418	6738	HULA-A	644	15	15
325	3310	HULA-A	675	45	15
412	6453	HULA-A	745	10	15
445	7314	HULA-A	770	30	15
452	7506	HULA-A	848	15	15
483	9443	HULA-A	879	45	15
71	3187	HULA-A	948	15	20
441	7310	HULA-A	979	25	15
72	3187	HULA-A	1048	15	9
888	DOWN	HULA-A	1063	61	0
414	6453	HULA-A	1140	30	15
469	8896	HULA-A	1186	45	15
416	6453	HULA-A	1247	30	15
434	7304	HULA-A	1293	15	15
392	6012	HULA-A	1389	35	15
470	9363	HULA-B	50	10	15
888	DOWN	HULA-B	64	90	0
2	286	HULA-B	174	16	20
134	9757	HULA-B	293	16	20
51	2532	HULA-B	359	14	20
52	2532	HULA-B	457	16	20
888	DOWN	HULA-B	725	180	0
92	4774	HULA-B	927	14	20
70	3187	HULA-B	948	15	7
150	9757	HULA-B	1037	16	20
58	2532	HULA-B	1200	15	20
512	7304	HULA-B	1262	10	15
35	1056	HULA-B	1328	14	20
37	1056	HULA-B	1425	15	20
100	6553	COOK-A	17	15	15
888	DOWN	COOK-A	32	150	0

417	6453	COOK-A	220	15	15
38	1132	COOK-A	315	14	20
504	7304	COOK-A	345	15	15
481	9442	COOK-A	390	10	15
413	6453	COOK-A	430	30	15
394	6071	COOK-A	476	20	15
315	2567	COOK-A	557	15	15
888	DOWN	COOK-A	577	70	0
395	6071	COOK-A	663	15	15
383	5953	COOK-A	697	20	15
428	7304	COOK-A	733	10	15
145	9757	COOK-A	832	15	20
888	DOWN	COOK-A	847	54	0
8	286	COOK-A	921	12	20
57	2532	COOK-A	996	14	15
328	3726	COOK-A	1031	195	10
331	3726	COOK-A	1237	165	10
382	5953	COOK-A	1418	20	15
77	4774	COOK-B	11	14	20
380	5953	COOK-B	41	20	15
479	9434	COOK-B	77	30	15
443	7314	COOK-B	123	15	15
312	2272	COOK-B	188	45	15
43	1748	COOK-B	327	16	20
503	7304	COOK-B	359	15	15
480	9441	COOK-B	390	10	15
44	1748	COOK-B	427	10	20
888	DOWN	COOK-B	437	62	0
457	7641	COOK-B	520	57	15
491	9783	COOK-B	662	91	15
446	7314	COOK-B	770	30	15
404	6392	COOK-B	816	30	15
147	9757	COOK-B	933	15	20
868	DOWN	COOK-B	948	61	0
327	3726	COOK-B	1031	50	10
466	8896	COOK-B	1097	40	15
32	1056	COOK-B	1223	16	15
301	470	COOK-B	1255	45	15
119	6553	COOK-B	1392	14	15
471	9363	COOK-B	1422	10	15
333	4035	INDI-A	10	10	10
78	4774	INDI-A	52	14	20
888	DOWN	INDI-A	66	52	0
131	9757	INDI-A	138	14	20
347	4845	INDI-A	168	30	15
326	3726	INDI-A	228	250	20
399	6280	INDI-A	494	30	15
360	5329	INDI-A	540	20	15
400	6280	INDI-A	640	30	15

888	DOWN	INDI-A	701	88	0
431	7304	INDI-A	805	205	15
453	7506	INDI-A	1026	10	15
433	7304	INDI-A	1052	95	15
888	DOWN	INDI-A	1151	111	0
33	1056	INDI-A	1282	16	15
370	5681	INDI-A	1314	45	15
322	3160	INDI-A	1375	20	15
488	9446	INDI-A	1411	25	15
398	6280	BOSS-A	30	20	15
352	4955	BOSS-A	71	10	20
393	6071	BOSS-A	190	10	15
359	5037	BOSS-A	220	20	15
17	1056	BOSS-A	290	14	20
888	DOWN	BOSS-A	304	65	0
19	1056	BOSS-A	389	17	20
303	712	BOSS-A	430	30	20
105	6553	BOSS-A	481	14	20
85	4774	BOSS-A	529	15	15
106	6553	BOSS-A	580	11	20
141	9757	BOSS-A	628	15	20
888	DOWN	BOSS-A	650	140	0
27	1056	BOSS-A	920	11	20
330	3726	BOSS-A	960	240	10
463	8275	BOSS-A	1216	30	15
888	DOWN	BOSS-A	1292	59	0
127	7050	BOSS-A	1358	14	20
308	2124	BOSS-A	1383	20	10
888	DOWN	BOSS-B	0	115	0
42	1748	BOSS-B	135	15	20
386	6012	BOSS-B	166	86	15
461	8275	BOSS-B	270	15	15
409	6451	BOSS-B	300	30	15
419	7225	BOSS-B	361	10	15
342	4373	BOSS-B	387	45	15
65	3187	BOSS-B	455	3	20
344	4524	BOSS-B	469	30	10
6	286	BOSS-B	514	12	15
501	7506	BOSS-B	542	15	15
7	286	BOSS-B	613	17	15
408	6451	BOSS-B	700	15	15
314	2524	BOSS-B	731	15	15
340	4035	BOSS-B	760	10	10
310	2124	BOSS-B	780	20	10
343	4524	BOSS-B	811	15	10
442	7310	BOSS-B	865	20	15
888	DOWN	BOSS-B	917	83	0
29	1056	BOSS-B	1019	17	20
304	712	BOSS-B	1057	30	20

30	1056	BOSS-B	1121	14	20
95	4774	BOSS-B	1191	14	16
74	3187	BOSS-B	1288	10	15
156	9757	BOSS-B	1311	10	13
454	7506	BOSS-B	1354	10	15
75	3187	BOSS-B	1386	17	15
888	DOWN	BOSS-B	1403	37	0
373	5775	LION-A	15	45	15
365	5329	LION-A	76	20	15
346	4832	LION-A	112	10	15
888	DOWN	LION-A	141	121	0
103	6553	LION-A	287	13	15
476	9366	LION-A	320	15	15
339	4035	LION-A	370	30	10
338	4035	LION-A	411	10	10
351	4955	LION-A	442	10	20
353	4955	LION-A	473	10	20
422	7225	LION-A	499	10	15
139	9757	LION-A	527	10	15
320	3055	LION-A	553	15	15
334	4035	LION-A	579	10	10
888	DOWN	LION-A	600	99	0
25	1056	LION-A	715	16	20
126	7050	LION-A	756	15	20
89	4774	LION-A	805	9	15
345	4524	LION-A	830	30	10
112	6553	LION-A	897	13	15
337	4035	LION-A	921	30	10
472	9364	LION-A	967	72	15
402	6391	LION-A	1060	30	15
888	DOWN	LION-A	1096	59	0
403	6391	LION-A	1171	10	15
475	9366	LION-A	1200	30	15
485	9444	LION-A	1290	10	15
363	5329	LION-A	1320	20	15
474	9366	LION-A	1356	15	15
349	4845	LION-B	15	30	15
14	1056	LION-B	89	17	20
888	DOWN	LION-B	109	59	0
102	6553	LION-B	188	15	20
80	4774	LION-B	233	14	20
135	9757	LION-B	323	15	15
348	4845	LION-B	354	30	15
136	9757	LION-B	424	16	20
888	DOWN	LION-B	440	80	0
427	7304	LION-B	536	10	15
459	7837	LION-B	562	25	15
24	1056	LION-B	615	11	20
388	6012	LION-B	697	35	15

440	7310	LION-B	743	10	10
366	5329	LION-B	770	20	15
26	1056	LION-B	816	15	20
888	DOWN	LION-B	838	71	0
484	9444	LION-B	925	30	15
335	4035	LION-B	968	10	10
114	6553	LION-B	994	15	15
143	9757	LION-B	1011	10	2
151	9757	LION-B	1108	16	20
391	6012	LION-B	1140	20	15
153	9757	LION-B	1210	15	15
387	6012	LION-B	1241	25	15
460	7837	LION-B	1283	35	15
98	5821	LION-B	1346	11	20
336	4035	LION-B	1368	10	10
36	1056	LION-B	1401	12	15
405	6392	LION-B	1429	10	15
482	9443	GUAM-A	15	10	15
379	5775	GUAM-A	40	20	15
888	DOWN	GUAM-A	141	62	0
16	1056	GUAM-A	223	15	20
500	3726	GUAM-A	254	45	15
104	6553	GUAM-A	319	13	15
321	3160	GUAM-A	350	20	15
82	4774	GUAM-A	397	15	15
502	7225	GUAM-A	428	15	15
506	2567	GUAM-A	459	15	15
137	9757	GUAM-A	494	14	15
140	9757	GUAM-A	594	15	15
465	8701	GUAM-A	350	65	15
377	5775	GUAM-A	731	20	15
888	DOWN	GUAM-A	753	68	0
509	7310	GUAM-A	842	10	15
329	3726	GUAM-A	900	40	10
401	6374	GUAM-A	956	15	15
28	1056	GUAM-A	987	13	15
511	7225	GUAM-A	1024	15	15
332	3726	GUAM-A	1050	32	10
319	3028	GUAM-A	1102	15	15
425	7225	GUAM-A	1133	10	15
520	6012	GUAM-A	1159	20	15
155	9757	GUAM-A	1241	16	20
160	9845	GUAM-A	1284	14	20
487	9445	GUAM-A	1314	10	15
158	9757	GUAM-A	1344	11	20
385	5953	GUAM-A	1371	30	15
372	5775	GUAM-A	1417	20	15
371	5775	GUAM-B	15	20	15
306	1864	GUAM-B	60	10	20

324	3160	GUAM-B	90	20	15
384	5953	GUAM-B	177	75	15
498	3726	GUAM-B	268	240	15
888	DOWN	GUAM-B	565	72	0
374	5775	GUAM-B	653	20	15
521	6012	GUAM-B	689	25	15
376	5775	GUAM-B	730	20	15
378	5775	GUAM-B	766	95	15
519	6012	GUAM-B	877	175	15
94	4774	GUAM-B	1125	15	15
888	DOWN	GUAM-B	1140	135	0
436	7304	GUAM-B	1291	10	15
316	2567	GUAM-B	1321	45	15
406	6394	GUAM-B	1390	10	15
397	6142	GUAM-B	1416	15	15
356	5037	PIKE-A	15	10	15
357	5037	PIKE-A	41	10	15
129	9757	PIKE-A	96	16	16
458	7837	PIKE-A	147	35	15
396	6142	PIKE-A	336	35	15
888	DOWN	PIKE-A	371	98	0
22	1056	PIKE-A	489	16	15
439	7310	PIKE-A	563	15	10
109	6553	PIKE-A	676	13	20
40	1132	PIKE-A	732	15	20
888	DOWN	PIKE-A	752	95	0
429	7304	PIKE-A	863	15	15
358	5037	PIKE-A	894	10	15
410	6451	PIKE-A	920	30	15
341	4373	PIKE-A	965	15	15
462	8275	PIKE-A	1100	30	15
411	6451	PIKE-A	1146	10	15
888	DOWN	PIKE-A	1201	77	0
117	6553	PIKE-A	1294	11	20
305	712	PIKE-A	1326	20	20
309	2124	PIKE-A	1360	20	10
96	4774	PIKE-A	1385	14	5
381	5953	PIKE-A	1415	20	15
494	4845	REEF-A	15	5	15
128	9757	REEF-A	40	11	20
888	DOWN	REEF-A	51	120	0
132	9757	REEF-A	138	16	15
124	7050	REEF-A	246	14	16
489	9446	REEF-A	300	25	15
125	7050	REEF-A	348	12	20
21	1056	REEF-A	430	15	15
448	7484	REEF-A	461	20	15
438	7310	REEF-A	492	10	10
888	DOWN	REEF-A	517	87	0

108 6553	REEF-A	619	14	15
369 5329	REEF-A	649	20	15
87 4774	REEF-A	686	14	15
110 6553	REEF-A	717	11	15
143 9757	REEF-A	789	16	20
364 5329	REEF-A	852	20	15
146 9757	REEF-A	891	14	15
492 9794	REEF-A	936	25	15
362 5329	REEF-A	980	45	15
367 5329	REEF-A	1041	55	15
455 7506	REEF-A	1112	10	1[illegible]
375 5775	REEF-A	1138	20	15
31 1056	REEF-A	1181	16	20
323 3160	REEF-A	1230	20	15
447 7484	REEF-A	1266	47	15
118 6553	REEF-A	1350	15	15
350 4845	REEF-A	1390	10	15
97 4774	REEF-A	1425	14	20

Unscheduled Supports

The following supports could not be scheduled.

Low Altitude Satellite Supports

2532

3187(2)

9757(2)

4774

Medium/High Altitude Satellite Supports

6012

1920

7225

7506

2124

5037

6012

7484

5037

5329

5329

9366

7304

7314

6012

7225

3726

7506

9365

6453

9783

5681

7304

3726

Appendix F. Schedule for Fourth Data Set

Overview

This appendix contains the following information.

- Schedule of Satellite supports and RTS downtime requirements
- List of unscheduled satellite supports

Schedule for data set 4

Note: The day was divided into one minute increments. The Strt Tme and Req Len columns are represented in minutes. For example, the start time of 1410 is equivelent to 2330. The request length of 90 is equivalent to one hour thirty minutes.

Sup	IRON	RTS	STrt Tme	Req Len	TAT
503	7304	POGO-A	15	15	15
12	1056	POGO-A	61	12	20
97	6553	POGO-A	135	14	15
99	6553	POGO-A	232	14	20
129	9757	POGO-A	268	15	20
19	1056	POGO-A	358	17	20
888	DOWN	POGO-A	385	20	0
102	6553	POGO-A	426	15	20
24	1056	POGO-A	561	17	15
427	7304	POGO-A	594	10	15
25	1056	POGO-A	664	16	15
420	7225	POGO-A	696	15	15
92	5821	POGO-A	731	13	20
27	1056	POGO-A	766	16	20
107	6553	POGO-A	823	11	15
116	6790	POGO-A	866	15	20
888	DOWN	POGO-A	881	76	0
142	9757	POGO-A	977	13	20
478	9434	POGO-A	1006	15	15
86	4774	POGO-A	1041	14	15
145	9757	POGO-A	1077	12	20
34	1056	POGO-A	1172	15	20
888	DOWN	POGO-A	1187	67	0
149	9757	POGO-A	1274	15	20
302	712	POGO-A	1320	15	20

44	1748	POGO-A	1369	13	20
94	5821	POGO-A	1407	13	20
495	7310	POGO-B	15	10	15
39	1132	POGO-B	65	11	20
888	DOWN	POGO-B	76	69	0
74	4774	POGO-B	163	10	15
307	1920	POGO-B	213	10	15
59	3187	POGO-B	283	16	9
3	286	POGO-B	324	12	20
76	4774	POGO-B	361	12	20
21	1056	POGO-B	459	17	20
43	1748	POGO-B	498	5	20
320	3055	POGO-B	519	15	15
78	4774	POGO-B	556	15	20
314	2524	POGO-B	587	15	15
104	6553	POGO-B	623	13	20
492	9794	POGO-B	652	25	15
388	6012	POGO-B	697	35	15
81	4774	POGO-B	751	14	16
64	3187	POGO-B	789	15	15
888	DOWN	POGO-B	810	190	0
390	6012	POGO-B	1016	110	15
513	470	POGO-B	1213	15	15
460	7837	POGO-B	1295	35	15
517	9783	POGO-B	1346	15	15
70	3187	POGO-B	1406	17	16
45	2532	POGO-C	33	15	10
479	9434	POGO-C	70	30	15
325	3310	POGO-C	116	45	15
114	6790	POGO-C	264	17	20
317	2941	POGO-C	310	15	15
419	7225	POGO-C	341	10	15
4	286	POGO-C	427	14	7
481	9442	POGO-C	457	10	15
62	3187	POGO-C	483	17	15
5	286	POGO-C	528	16	15
133	9757	POGO-C	574	17	20
49	2532	POGO-C	645	12	15
115	6790	POGO-C	664	16	5
50	2532	POGO-C	745	14	10
51	2532	POGO-C	845	15	15
65	3187	POGO-C	894	12	15
52	2532	POGO-C	945	16	15
453	7506	POGO-C	1003	10	15
8	286	POGO-C	1032	17	15
888	DOWN	POGO-C	1049	61	0
9	286	POGO-C	1133	17	15
391	6012	POGO-C	1166	20	15
68	3187	POGO-C	1203	14	15

55	2532	POGO-C	1242	16	5
434	7304	POGO-C	1274	15	15
69	3187	POGO-C	1305	15	14
56	2532	POGO-C	1341	16	15
888	DOWN	POGO-C	1363	77	0
507	6012	HULA-A	15	180	15
888	DOWN	HULA-A	200	99	0
319	3028	HULA-A	315	15	15
90	5821	HULA-A	419	11	20
312	2272	HULA-A	446	45	15
91	5821	HULA-A	515	11	20
888	DOWN	HULA-A	527	112	0
408	6451	HULA-A	700	15	15
383	5953	HULA-A	731	20	15
445	7314	HULA-A	770	30	15
446	7314	HULA-A	816	30	15
888	DOWN	HULA-A	901	73	0
143	9757	HULA-A	994	15	19
153	9845	HULA-A	1032	15	20
67	3187	HULA-A	1079	12	20
411	6451	HULA-A	1107	10	15
416	6453	HULA-A	1215	30	15
512	7304	HULA-A	1261	10	15
471	9363	HULA-A	1287	10	15
124	7050	HULA-A	1377	13	17
520	6012	HULA-A	1406	20	15
888	DOWN	HULA-B	0	100	0
73	4774	HULA-B	142	15	20
2	286	HULA-B	199	17	20
417	6453	HULA-B	232	15	15
450	7506	HULA-B	263	10	15
318	3028	HULA-B	289	45	15
505	6142	HULA-B	350	15	15
508	6012	HULA-B	381	15	15
47	2532	HULA-B	419	17	20
888	DOWN	HULA-B	425	77	0
48	2532	HULA-B	522	11	20
316	2567	HULA-B	589	45	15
466	8896	HULA-B	650	40	15
401	6374	HULA-B	769	15	15
106	6553	HULA-B	804	14	20
83	4774	HULA-B	866	14	15
108	6553	HULA-B	902	13	20
71	3726	HULA-B	941	10	20
66	3187	HULA-B	976	17	20
441	7310	HULA-B	1009	25	15
43	1748	HULA-B	1067	15	20
888	DOWN	HULA-B	1085	114	0
123	7050	HULA-B	1275	15	20

464	8639	HULA-B	1306	15	15
381	5953	HULA-B	1342	20	15
392	6012	HULA-B	1389	35	15
380	5953	COOK-A	15	20	15
96	6553	COOK-A	52	13	15
888	DOWN	COOK-A	131	76	0
119	7050	COOK-A	227	14	20
313	2272	COOK-A	257	15	15
40	1132	COOK-A	349	14	20
396	6142	COOK-A	379	35	15
20	1056	COOK-A	452	9	20
394	6071	COOK-A	477	20	15
457	7641	COOK-A	520	57	15
395	6071	COOK-A	600	15	15
414	6453	COOK-A	630	30	15
888	DOWN	COOK-A	728	74	0
152	984[illegible]	COOK-A	834	16	20
140	9757	COOK-A	889	16	20
410	6451	COOK-A	921	30	15
888	DOWN	COOK-A	965	60	0
54	2532	COOK-A	1058	15	30
327	372[illegible]	COOK-A	1104	50	10
455	7506	COOK-A	1170	10	15
487	9445	COOK-A	1196	10	15
37	1056	COOK-A	1284	16	15
385	5953	COOK-A	1320	30	15
454	7506	COOK-A	1366	10	15
72	4774	COOK-B	48	13	20
412	6453	COOK-B	90	10	15
458	7837	COOK-B	116	35	15
384	5953	COOK-B	170	75	15
461	8275	COOK-B	270	15	15
409	6451	COOK-B	300	30	15
439	7310	COOK-B	342	15	10
504	7304	COOK-B	373	15	15
888	DOWN	COOK-B	454	118	0
472	9364	COOK-B	600	72	15
888	DOWN	COOK-B	734	89	0
433	7304	COOK-B	839	95	15
331	3726	COOK-B	960	165	10
888	DOWN	COOK-B	1126	113	0
468	8896	COOK-B	1255	40	15
521	6012	COOK-B	1311	25	15
113	6553	COOK-B	1426	14	20
95	6553	INDI-A	11	15	16
888	DOWN	INDI-A	26	120	0
324	3160	INDI-A	162	20	15
497	3726	INDI-A	198	55	15

338	4035	INDI-A	264	10	10
376	5775	INDI-A	290	20	15
348	4845	INDI-A	326	30	15
351	4955	INDI-A	390	10	20
399	6280	INDI-A	440	30	15
22	1056	INDI-A	491	12	20
345	4524	INDI-A	514	30	10
431	7304	INDI-A	560	205	15
377	5775	INDI-A	781	20	15
343	4524	INDI-A	812	15	10
888	DOWN	INDI-A	831	97	0
141	9757	INDI-A	948	16	20
42	1132	INDI-A	1026	11	20
375	5775	INDI-A	1053	20	15
888	DOWN	INDI-A	1073	83	0
488	9446	INDI-A	1200	25	15
322	3160	INDI-A	1241	20	15
516	5681	INDI-A	1277	15	15
38	1056	INDI-A	1344	13	20
112	6553	INDI-A	1387	12	12
405	6392	INDI-A	1415	10	15
386	6012	BOSS-A	27	86	15
470	9363	BOSS-A	129	10	15
501	7506	BOSS-A	155	15	15
15	1056	BOSS-A	252	10	15
888	DOWN	BOSS-A	262	67	0
18	1056	BOSS-A	349	16	15
480	9441	BOSS-A	390	10	15
101	6553	BOSS-A	419	12	15
438	7310	BOSS-A	442	10	10
77	4774	BOSS-A	469	12	15
353	4955	BOSS-A	502	10	20
134	9757	BOSS-A	584	13	15
41	1132	BOSS-A	667	15	20
888	DOWN	BOSS-A	694	77	0
340	4035	BOSS-A	782	10	10
404	6392	BOSS-A	810	30	15
370	5681	BOSS-A	856	45	15
335	4035	BOSS-A	912	10	10
328	3726	BOSS-A	993	195	10
469	8896	BOSS-A	1204	45	15
387	6012	BOSS-A	1265	25	15
888	DOWN	BOSS-A	1314	86	0
474	9366	BOSS-A	1416	15	15
306	1864	BOSS-B	20	10	20
125	9757	BOSS-B	54	16	15
888	DOWN	BOSS-B	70	34	0
46	2532	BOSS-B	124	15	5
393	6071	BOSS-B	190	10	15

397	6142	BOSS-B	279	15	15
341	4373	BOSS-B	318	15	15
423	7225	BOSS-B	349	105	15
61	3187	BOSS-B	480	9	20
888	DOWN	BOSS-B	489	82	0
63	3187	BOSS-B	576	16	15
315	2567	BOSS-B	608	15	15
6	286	BOSS-B	639	17	14
136	9757	BOSS-B	685	16	15
888	DOWN	BOSS-B	701	99	0
452	7506	BOSS-B	817	15	15
310	2124	BOSS-B	843	20	10
31	1056	BOSS-B	979	15	20
337	4035	BOSS-B	1005	30	10
477	9366	BOSS-B	1051	10	15
463	8275	BOSS-B	1100	30	15
510	7506	BOSS-B	1146	15	15
87	4774	BOSS-B	1226	15	15
888	DOWN	BOSS-B	1241	70	0
11	286	BOSS-B	1326	16	15
151	9757	BOSS-B	1366	14	20
309	2124	BOSS-B	1391	20	10
57	2532	BOSS-B	1426	14	15
888	DOWN	LION-A	0	100	0
361	5329	LION-A	116	20	15
359	5037	LION-A	220	20	15
347	4845	LION-A	256	30	15
437	7310	LION-A	320	20	10
339	4035	LION-A	350	30	10
342	4373	LION-A	413	45	15
422	7225	LION-A	474	10	15
426	7304	LION-A	504	15	15
888	DOWN	LION-A	524	77	0
26	1056	LION-A	675	14	20
428	7304	LION-A	705	10	15
28	1056	LION-A	776	16	15
429	7304	LION-A	808	15	15
82	4774	LION-A	839	12	15
29	1056	LION-A	878	12	20
888	DOWN	LION-A	890	79	0
442	7310	LION-A	985	20	15
85	4774	LION-A	1033	13	13
144	9757	LION-A	1066	13	15
402	6391	LION-A	1095	30	15
147	9757	LION-A	1165	16	20
403	6391	LION-A	1197	10	15
148	9757	LION-A	1268	14	15
462	8275	LION-A	1298	30	15
363	5329	LION-A	1344	20	15

349	4845	LION-B	15	30	15
373	5775	LION-B	61	45	15
365	5329	LION-B	122	20	15
98	6553	LION-B	223	14	15
75	4774	LION-B	269	15	14
888	DOWN	LION-B	274	86	0
131	9757	LION-B	380	16	20
476	9366	LION-B	412	15	15
493	9794	LION-B	443	15	15
132	9757	LION-B	482	14	20
344	4524	LION-B	507	30	10
360	5329	LION-B	553	20	15
334	4035	LION-B	584	10	10
369	5329	LION-B	612	20	15
888	DOWN	LION-B	677	146	0
368	5329	LION-B	839	15	15
122	7050	LION-B	898	15	20
109	6553	LION-B	931	14	15
362	5329	LION-B	980	45	15
367	5329	LION-B	1041	55	15
475	9366	LION-B	1200	30	15
888	DOWN	LION-B	1242	58	0
305	712	LION-B	1321	20	20
372	5775	LION-B	1360	20	15
350	4845	LION-B	1396	10	15
336	4035	LION-B	1417	10	10
379	5775	GUAM-A	15	20	15
482	9443	GUAM-A	51	10	15
888	DOWN	GUAM-A	82	80	0
14	1056	GUAM-A	182	16	20
500	3726	GUAM-A	214	45	15
17	1056	GUAM-A	285	12	15
321	3160	GUAM-A	313	20	15
100	6553	GUAM-A	352	15	15
486	9445	GUAM-A	390	30	15
413	6453	GUAM-A	436	30	15
473	9365	GUAM-A	482	73	15
888	DOWN	GUAM-A	560	73	0
135	9757	GUAM-A	653	11	20
459	7837	GUAM-A	680	25	15
120	7050	GUAM-A	726	15	16
378	5775	GUAM-A	757	95	15
30	1056	GUAM-A	945	16	20
332	3726	GUAM-A	972	32	10
301	470	GUAM-A	1023	45	15
483	9443	GUAM-A	1084	45	15
511	7225	GUAM-A	1145	15	15
425	7225	GUAM-A	1176	10	15
888	DOWN	GUAM-A	1190	57	0
509	7310	GUAM-A	1263	10	15

382	5953	GUAM-A	1350	20	15
371	5775	GUAM-B	15	20	15
490	9521	GUAM-B	51	90	15
502	7225	GUAM-B	157	15	15
506	2567	GUAM-B	188	15	15
888	DOWN	GUAM-B	271	150	0
491	9783	GUAM-B	437	91	15
407	6394	GUAM-B	544	30	15
374	5775	GUAM-B	590	20	15
415	6453	GUAM-B	626	115	15
888	DOWN	GUAM-B	793	72	0
329	3726	GUAM-B	900	40	10
330	3726	GUAM-B	960	240	10
323	3160	GUAM-B	1216	20	15
436	7304	GUAM-B	1252	10	15
150	9757	GUAM-B	1298	16	20
514	7304	GUAM-B	1330	15	15
406	6394	GUAM-B	1360	10	15
888	DOWN	GUAM-B	1391	49	0
356	5037	PIKE-A	15	10	15
443	7314	PIKE-A	40	15	15
357	5037	PIKE-A	71	10	15
127	9757	PIKE-A	153	17	13
888	DOWN	PIKE-A	180	150	0
303	712	PIKE-A	430	30	20
311	2124	PIKE-A	471	20	10
418	6738	PIKE-A	507	15	15
23	1056	PIKE-A	550	15	20
358	5037	PIKE-A	581	10	15
103	6553	PIKE-A	612	14	15
440	7310	PIKE-A	637	10	10
80	4774	PIKE-A	665	14	15
105	6553	PIKE-A	711	12	15
138	9757	PIKE-A	787	16	15
888	DOWN	PIKE-A	803	60	0
432	7304	PIKE-A	879	130	15
304	712	PIKE-A	1030	30	20
32	1056	PIKE-A	1082	12	20
467	8896	PIKE-A	1110	25	15
308	2124	PIKE-A	1146	20	10
35	1056	PIKE-A	1182	16	15
888	DOWN	PIKE-A	1198	72	0
111	6553	PIKE-A	1327	14	20
518	1920	PIKE-A	1357	45	15
89	4774	PIKE-A	1422	13	20
888	DOWN	REEF-A	0	90	0
126	9757	REEF-A	94	16	20
449	7506	REEF-A	126	49	15

128 9757	REEF-A	196	14	20
326 3726	REEF-A	231	250	20
888 DOWN	REEF-A	501	105	0
79 4774	REEF-A	626	12	15
465 8701	REEF-A	654	65	15
366 5329	REEF-A	735	20	15
139 9757	REEF-A	846	16	20
364 5329	REEF-A	878	20	15
484 9444	REEF-A	914	30	15
447 7484	REEF-A	960	47	15
33 1056	REEF-A	1142	13	20
888 DOWN	REEF-A	1153	69	0
36 1056	REEF-A	1242	15	20
485 9444	REEF-A	1273	10	15
88 4774	REEF-A	1364	14	20
489 9446	REEF-A	1394	25	15

Unscheduled Supports

The following supports could not be scheduled.

Low Altitude Satellite Supports

1056

3187

9757

6790

7050

9845

Medium/High Altitude Satellite Supports

6012

7225

3726

5037

7484

6280

5037

6012

7314

7304

7484

6280

3726

4845

4955

4832

4035

Appendix G. Schedule for Fifth Data Set

Overview

This appendix contains the following information.

- Schedule of Satellite supports and RTS downtime requirements
- List of unscheduled satellite supports

Schedule for data set 5

Note: The day was divided into one minute increments. The Strt Tme and Req Len columns are represented in minutes. For example, the start time of 1410 is equivelent to 2330. The request length of 90 is equivalent to one hour thirty minutes.

Sup No. Ident GTS Strt Tme (hhmm) Sup Len (hhmm) TAT

Sup	IRON	RTS	Strt Tme	Req Len	TAT
101	6553	POGO-A	102	14	15
888	DOWN	POGO-A	116	94	0
79	4774	POGO-A	225	11	12
81	4774	POGO-A	324	12	15
20	1056	POGO-A	399	16	20
888	DOWN	POGO-A	415	65	0
23	1056	POGO-A	500	16	20
107	6553	POGO-A	588	14	15
504	7225	POGO-A	626	70	15
142	9757	POGO-A	719	16	20
90	4774	POGO-A	812	14	20
91	4774	POGO-A	908	14	15
888	DOWN	POGO-A	930	70	0
439	7506	POGO-A	1016	10	15
312	2272	POGO-A	1042	15	15
115	6553	POGO-A	1086	13	20
152	9757	POGO-A	1119	13	15
46	1748	POGO-A	1222	12	17
440	7506	POGO-A	1250	10	15
116	6553	POGO-A	1282	14	15
157	9757	POGO-A	1317	16	15
888	DOWN	POGO-A	1333	60	0

48	1748	POGO-A	1413	14	20
306	1864	POGO-B	20	10	20
130	9757	POGO-B	107	17	20
888	DOWN	POGO-B	120	75	0
133	9757	POGO-B	210	16	15
402	6738	POGO-B	268	15	15
18	1056	POGO-B	298	16	15
313	2524	POGO-B	347	15	15
98	5821	POGO-B	382	9	20
83	4774	POGO-B	422	14	20
389	6392	POGO-B	452	10	15
314	2567	POGO-B	478	15	15
138	9757	POGO-B	517	16	20
888	DOWN	POGO-B	533	64	0
86	4774	POGO-B	617	15	20
451	8639	POGO-B	648	15	15
54	2532	POGO-B	682	14	15
888	DOWN	POGO-B	710	150	0
111	6553	POGO-B	888	11	20
122	6790	POGO-B	924	14	15
113	6553	POGO-B	988	11	15
149	9757	POGO-B	1020	13	15
93	4774	POGO-B	1101	15	15
45	1748	POGO-B	1126	11	10
154	9757	POGO-B	1218	14	20
422	7304	POGO-B	1248	45	15
34	1056	POGO-B	1312	13	16
99	5821	POGO-B	1353	14	20
412	7225	POGO-B	1383	50	15
888	DOWN	POGO-C	0	50	0
49	2532	POGO-C	70	16	5
62	3187	POGO-C	153	17	15
316	3028	POGO-C	186	15	15
63	3187	POGO-C	254	16	15
4	286	POGO-C	298	12	15
64	3187	POGO-C	354	16	15
5	286	POGO-C	401	13	15
66	3187	POGO-C	454	17	15
301	470	POGO-C	487	15	15
84	4774	POGO-C	520	14	15
67	3187	POGO-C	555	17	15
53	2532	POGO-C	581	12	5
120	6790	POGO-C	620	16	15
888	DOWN	POGO-C	636	77	0
474	9521	POGO-C	729	15	15
55	2532	POGO-C	782	15	5
121	6790	POGO-C	822	15	15
69	3187	POGO-C	863	13	15
56	2532	POGO-C	882	16	5

405	7225	POGO-C	914	15	15
9	286	POGO-C	1007	16	15
491	7506	POGO-C	1039	15	15
410	7225	POGO-C	1070	15	15
10	286	POGO-C	1107	17	15
409	7225	POGO-C	1140	10	15
73	3187	POGO-C	1174	13	15
11	286	POGO-C	1208	17	15
123	6790	POGO-C	1234	12	9
59	2532	POGO-C	1279	16	15
12	286	POGO-C	1310	16	15
60	2532	POGO-C	1379	16	15
13	286	POGO-C	1413	15	8
888	DOWN	HULA-A	0	90	0
380	6071	HULA-A	106	30	15
484	6012	HULA-A	152	95	15
3	286	HULA-A	275	15	15
435	7506	HULA-A	306	15	15
481	9794	HULA-A	359	15	15
39	1132	HULA-A	408	15	20
466	9443	HULA-A	439	25	15
888	DOWN	HULA-A	470	70	0
436	7506	HULA-A	556	250	15
429	7310	HULA-A	822	20	15
71	3187	HULA-A	948	15	20
888	DOWN	HULA-A	963	65	0
72	3187	HULA-A	1048	15	9
414	7225	HULA-A	1079	130	15
494	7304	HULA-A	1225	15	15
495	6142	HULA-A	1303	95	15
427	7310	HULA-A	1414	20	15
456	9363	HULA-B	30	56	15
381	6071	HULA-B	102	30	15
2	286	HULA-B	174	16	20
437	7506	HULA-B	259	10	15
134	9757	HULA-B	293	16	20
51	2532	HULA-B	359	14	20
367	5953	HULA-B	389	20	15
52	2532	HULA-B	457	16	20
888	DOWN	HULA-B	473	93	0
454	8896	HULA-B	601	40	15
475	9783	HULA-B	660	15	15
501	6542	HULA-B	696	110	20
888	DOWN	HULA-B	831	76	0
92	4774	HULA-B	927	14	20
70	3187	HULA-B	948	15	7
150	9757	HULA-B	1037	16	20
411	7225	HULA-B	1069	65	15
58	2532	HULA-B	1200	15	20

888	DOWN	HULA-B	1215	93	0
35	1056	HULA-B	1328	1[illegible]	20
470	9445	HULA-B	1358	10	15
37	1056	HULA-B	1425	15	20
100	6553	COOK-A	17	15	15
888	DOWN	COOK-A	52	69	0
485	6012	COOK-A	137	130	15
38	1132	COOK-A	315	14	20
888	DOWN	COOK-A	329	60	0
407	7225	COOK-A	405	260	15
415	7304	COOK-A	718	15	15
888	DOWN	COOK-A	734	66	0
145	9757	COOK-A	832	15	20
393	6451	COOK-A	870	30	15
8	286	COOK-A	921	12	20
417	7304	COOK-A	949	10	15
57	2532	COOK-A	996	14	15
450	8275	COOK-A	1026	15	15
888	DOWN	COOK-A	1049	71	0
441	7506	COOK-A	1154	120	15
446	7837	COOK-A	1289	35	15
391	6394	COOK-A	1340	30	15
378	6012	COOK-A	1400	35	15
77	4774	COOK-B	11	14	20
464	9442	COOK-B	41	64	15
430	7314	COOK-B	121	15	15
461	9434	COOK-B	152	30	15
888	DOWN	COOK-B	240	67	0
43	1748	COOK-B	327	16	20
315	2941	COOK-B	361	15	15
404	7225	COOK-B	393	10	15
44	1748	COOK-B	427	10	20
507	7225	COOK-B	453	260	15
420	7304	COOK-B	748	10	15
399	6453	COOK-B	780	10	15
888	DOWN	COOK-B	799	110	0
147	9757	COOK-B	933	15	20
323	3726	COOK-B	1023	23	20
396	6451	COOK-B	1062	30	15
433	7314	COOK-B	1120	20	15
888	DOWN	COOK-B	1141	67	0
32	1056	COOK-B	1223	16	15
443	7506	COOK-B	1255	110	15
119	6553	COOK-B	1392	14	15
307	1920	COOK-B	1422	15	15
343	4845	INDI-A	15	10	15
78	4774	INDI-A	52	14	20
888	DOWN	INDI-A	66	50	0

131 9757	INDI-A	138	14	20
322 3726	INDI-A	222	22	20
472 9446	INDI-A	255	35	10
387 6391	INDI-A	330	30	15
455 8896	INDI-A	376	45	15
408 7225	INDI-A	475	10	15
452 8701	INDI-A	500	52	15
418 7304	INDI-A	578	70	15
478 7506	INDI-A	664	15	15
421 7304	INDI-A	698	165	15
508 7225	INDI-A	879	60	15
489 3055	INDI-A	960	85	15
888 DOWN	INDI-A	1120	142	0
33 1056	INDI-A	1282	16	15
319 3160	INDI-A	1320	20	15
338 4524	INDI-A	1356	15	15
487 3726	INDI-A	1387	50	15
444 7641	BOSS-A	15	44	15
303 712	BOSS-A	75	30	15
395 6451	BOSS-A	121	10	15
371 6012	BOSS-A	147	71	15
333 4035	BOSS-A	223	30	5
17 1056	BOSS-A	290	14	20
888 DOWN	BOSS-A	304	65	0
19 1056	BOSS-A	389	17	20
350 5037	BOSS-A	413	15	6
105 6553	BOSS-A	481	14	20
85 4774	BOSS-A	529	15	15
888 DOWN	BOSS-A	545	63	0
106 6553	BOSS-A	580	11	20
141 9757	BOSS-A	628	15	20
372 6012	BOSS-A	679	35	15
310 2124	BOSS-A	725	20	10
388 6391	BOSS-A	790	10	15
334 4035	BOSS-A	863	10	5
486 9363	BOSS-A	889	10	15
27 1056	BOSS-A	920	11	20
888 DOWN	BOSS-A	931	69	0
325 3726	BOSS-A	1021	255	20
460 9306	BOSS-A	1292	10	15
493 7304	BOSS-A	1318	20	15
127 7050	BOSS-A	1358	14	20
888 DOWN	BOSS-A	1372	68	0
390 6392	BOSS-B	15	30	15
463 9441	BOSS-B	61	10	15
42 1748	BOSS-B	135	15	20
337 4524	BOSS-B	166	30	15
888 DOWN	BOSS-B	200	70	0
384 6280	BOSS-B	286	10	15

476	9794	BCSS-B	315	30	15
888	DOWN	BOSS-B	350	90	0
65	3187	BOSS-B	455	3	20
6	286	BOSS-B	514	12	15
302	470	BOSS-B	542	45	15
7	286	BOSS-B	613	17	15
888	DOWN	BOSS-B	640	55	0
374	6012	BOSS-B	711	240	15
29	1056	BOSS-B	1019	17	20
888	DOWN	BOSS-B	1046	55	0
30	1056	BOSS-B	1121	14	20
394	6451	BOSS-B	1151	15	15
95	4774	BOSS-B	1191	14	16
459	9366	BOSS-B	1221	10	15
375	6012	BOSS-B	1247	10	15
74	3187	BOSS-B	1288	10	15
156	9757	BOSS-B	1311	10	13
428	7310	BOSS-B	1355	10	15
75	3187	BOSS-B	1386	17	15
349	5037	BCSS-B	1410	10	6
332	4035	BOSS-B	1426	10	5
354	5329	LION-A	15	20	15
888	DOWN	LION-A	55	150	0
311	2124	LION-A	216	20	10
103	6553	LION-A	287	13	15
505	7225	LION-A	316	75	15
385	6280	LION-A	430	30	15
345	4955	LION-A	500	10	15
139	9757	LION-A	527	10	15
888	DOWN	LION-A	537	63	0
360	5775	LION-A	616	20	15
25	1056	LION-A	715	16	20
126	7050	LION-A	756	15	20
89	4774	LION-A	805	9	15
377	6012	LION-A	830	25	15
112	6553	LION-A	897	13	15
468	9444	LION-A	975	10	15
364	5775	LION-A	1000	80	15
373	6012	LION-A	1096	70	15
888	DOWN	LION-A	1202	80	0
365	5775	LION-A	1320	40	15
383	6280	LION-A	1376	64	15
473	9446	LION-B	10	25	10
309	2124	LION-B	46	20	10
14	1056	LION-B	89	17	20
344	4845	LION-B	122	30	15
329	4035	LION-B	158	10	5
102	6553	LION-B	188	15	20
80	4774	LION-B	233	14	20

888 DOWN	LION-B	247	61	0
135 9757	LION-B	323	15	15
359 5775	LION-B	354	20	15
327 4035	LION-B	380	10	5
136 9757	LION-B	424	16	20
416 7304	LION-B	494	45	15
362 5775	LION-B	555	40	15
24 1056	LION-B	615	11	20
888 DOWN	LION-B	626	90	0
353 5329	LION-B	732	50	15
26 1056	LION-B	816	15	20
376 6012	LION-B	847	130	15
114 6553	LION-B	994	15	15
148 9757	LION-B	1011	10	2
331 4035	LION-B	1027	10	5
330 4035	LION-B	1043	10	5
151 9757	LION-B	1108	16	20
339 4524	LION-B	1140	30	15
153 9757	LION-B	1210	15	15
888 DOWN	LION-B	1225	100	0
98 5821	LION-B	1346	11	20
36 1056	LION-B	1401	12	15
888 DOWN	GUAM-A	0	90	0
387 5681	GUAM-A	106	35	15
403 6738	GUAM-A	157	45	15
16 1056	GUAM-A	223	15	20
503 6738	GUAM-A	254	45	15
104 6553	GUAM-A	319	13	15
386 6374	GUAM-A	363	15	15
82 4774	GUAM-A	397	15	15
888 DOWN	GUAM-A	412	67	0
137 9757	GUAM-A	494	14	15
335 4373	GUAM-A	524	15	15
397 6453	GUAM-A	555	15	15
140 9757	GUAM-A	594	15	15
888 DOWN	GUAM-A	610	90	0
368 5953	GUAM-A	880	20	15
28 1056	GUAM-A	987	13	15
492 7225	GUAM-A	1016	15	15
509 7225	GUAM-A	1047	60	15
888 DOWN	GUAM-A	1111	110	0
155 9757	GUAM-A	1241	16	20
160 9845	GUAM-A	1284	14	20
158 9757	GUAM-A	1344	11	20
499 6012	GUAM-A	1371	25	15
318 3160	GUAM-B	15	20	15
469 9445	GUAM-B	51	30	15
479 7225	GUAM-B	97	100	15
888 DOWN	GUAM-B	211	89	0

502 6738	GUAM-B	316	115	15
370 5953	GUAM-B	447	20	15
317 3055	GUAM-B	489	45	15
358 5681	GUAM-B	550	15	15
320 3160	GUAM-B	600	20	15
480 7225	GUAM-B	636	15	15
392 6394	GUAM-B	667	10	15
888 DOWN	GUAM-B	720	90	0
398 6453	GUAM-B	826	10	15
369 5953	GUAM-B	880	20	15
465 9443	GUAM-B	916	25	15
490 3726	GUAM-B	960	40	15
888 DOWN	GUAM-B	1020	90	0
94 4774	GUAM-B	1125	15	15
488 3726	GUAM-B	1156	240	15
366 5775	GUAM-B	1412	20	15
888 DOWN	PIKE-A	0	70	0
129 9757	PIKE-A	96	16	16
445 7837	PIKE-A	134	35	15
382 6142	PIKE-A	185	82	15
448 8275	PIKE-A	340	30	15
449 8275	PIKE-A	386	30	15
22 1056	PIKE-A	489	16	15
305 712	PIKE-A	521	30	15
888 DOWN	PIKE-A	560	96	0
109 6553	PIKE-A	676	13	20
40 1132	PIKE-A	732	15	20
447 7837	PIKE-A	782	30	15
321 3310	PIKE-A	828	45	15
438 7506	PIKE-A	889	20	15
308 2124	PIKE-A	920	20	10
888 DOWN	PIKE-A	950	50	0
453 8896	PIKE-A	1029	40	15
336 4373	PIKE-A	1085	45	15
348 5037	PIKE-A	1137	10	6
413 7225	PIKE-A	1168	65	15
117 6553	PIKE-A	1294	11	20
888 DOWN	PIKE-A	1305	60	0
96 4774	PIKE-A	1385	14	5
304 712	PIKE-A	1415	15	15
128 9757	REEF-A	40	11	20
434 7506	REEF-A	86	34	15
132 9757	REEF-A	138	16	15
888 DOWN	REEF-A	154	76	0
124 7050	REEF-A	246	14	16
125 7050	REEF-A	348	12	20
426 7310	REEF-A	403	10	15
21 1056	REEF-A	430	15	15
888 DOWN	REEF-A	470	70	0

351	5329	REEF-A	556	20	15
108	6553	REEF-A	619	14	15
87	4774	REEF-A	686	14	15
110	6553	REEF-A	717	11	15
425	7310	REEF-A	744	20	15
143	9757	REEF-A	789	16	20
888	DOWN	REEF-A	805	71	0
146	9757	REEF-A	891	14	15
341	4845	REEF-A	982	30	15
352	5329	REEF-A	1028	20	15
361	5775	REEF-A	1109	20	15
477	9794	REEF-A	1145	15	15
31	1056	REEF-A	1181	16	20
458	9365	REEF-A	1213	58	15
471	9446	REEF-A	1282	50	10
118	6553	REEF-A	1350	15	15
356	5329	REEF-A	1381	20	15
97	4774	REEF-A	1425	14	20

Unscheduled Supports

The following supports could not be scheduled.

Low Altitude Satellite Supports

1132

2532

3187(2)

9757(2)

4774

Medium/High Altitude Satellite Supports

7304

7506

5037

3726

1920

9364

5037

7314

7314

6071

6542

6542

7304

7304

7225

7225

5329

6012

7304

5775

7225

9444

3055

9434

4035

4832

7310

4845

Appendix H. Schedule for Sixth Data Set

Overview

This appendix contains the following information.

- Schedule of Satellite supports and RTS downtime requirements
- List of unscheduled satellite supports

Schedule for data set 6

Note: The day was divided into one minute increments. The Strt Tme and Req Len columns are represented in minutes. For example, the start time of 1410 is equivelent to 2330. The request length of 90 is equive

Sup	IRON	RTS	Strt Tme	Req Len	TAT
390	6392	POGO-A	15	30	15
888	DOWN	POGO-A	190	115	0
131	9757	POGO-A	325	16	20
445	7837	POGO-A	357	35	15
313	2524	POGO-A	408	15	15
301	470	POGO-A	487	15	15
20	1056	POGO-A	521	16	15
455	8896	POGO-A	553	45	15
22	1056	POGO-A	623	16	20
425	7310	POGO-A	655	20	15
504	7225	POGO-A	691	70	15
88	4774	POGO-A	787	14	18
26	1056	POGO-A	827	16	20
90	4774	POGO-A	884	14	15
29	1056	POGO-A	929	16	15
888	DOWN	POGO-A	1021	92	0
147	9757	POGO-A	1133	12	15
443	7506	POGO-A	1161	110	15
375	6012	POGO-A	1287	10	15
888	DOWN	POGO-A	1310	90	0
54	1748	POGO-A	1420	15	20
41	1132	POGO-B	3	6	20
888	DOWN	POGO-B	26	176	0
358	5681	POGO-B	218	15	15

435	7506	POGO-B	278	15	15
508	7225	POGO-B	326	60	15
17	1056	POGO-B	419	16	15
102	6553	POGO-B	460	15	15
52	1748	POGO-B	551	6	20
888	DOWN	POGO-B	556	85	0
420	7304	POGO-B	659	10	15
86	4774	POGO-B	690	15	20
401	6542	POGO-B	800	10	20
377	6012	POGO-B	826	25	15
438	7506	POGO-B	867	20	15
501	6542	POGO-B	908	110	20
144	9757	POGO-B	1034	12	15
491	7506	POGO-B	1062	15	15
410	7225	POGO-B	1093	15	15
33	1056	POGO-B	1132	15	20
888	DOWN	POGO-B	1147	70	0
36	1056	POGO-B	1232	14	20
151	9757	POGO-B	1331	15	20
509	7225	POGO-B	1362	60	15
357	5681	POGO-C	110	35	15
402	6738	POGO-C	268	15	15
116	6790	POGO-C	308	16	20
888	DOWN	POGO-C	322	116	0
3	286	POGO-C	453	14	15
389	6392	POGO-C	483	10	15
69	3187	POGO-C	512	17	15
4	286	POGO-C	554	16	15
59	2532	POGO-C	607	12	15
5	286	POGO-C	656	16	15
60	2532	POGO-C	707	14	15
888	DOWN	POGO-C	721	66	0
61	2532	POGO-C	808	15	15
109	6553	POGO-C	858	10	15
62	2532	POGO-C	907	16	15
6	286	POGO-C	958	16	20
117	6790	POGO-C	1015	12	15
7	286	POGO-C	1058	17	15
417	7304	POGO-C	1091	10	15
73	3187	POGO-C	1131	12	15
8	286	POGO-C	1159	16	15
64	2532	POGO-C	1205	16	15
74	3187	POGO-C	1233	14	12
9	286	POGO-C	1260	16	5
65	2532	POGO-C	1304	16	4
75	3187	POGO-C	1334	16	5
10	286	POGO-C	1362	16	12
66	2532	POGO-C	1404	16	14
12	1056	HULA-A	40	14	20

888	DOWN	HULA-A	54	151	0
2	286	HULA-A	225	16	20
367	5953	HULA-A	320	20	15
481	9794	HULA-A	359	15	15
133	9757	HULA-A	408	15	18
466	9443	HULA-A	439	25	15
431	7314	HULA-A	567	30	15
24	1056	HULA-A	706	16	20
888	DOWN	HULA-A	733	71	0
393	6451	HULA-A	870	30	15
465	9443	HULA-A	916	25	15
396	6451	HULA-A	1000	30	15
145	9757	HULA-A	1050	17	20
46	1132	HULA-A	1101	15	20
413	7225	HULA-A	1132	65	15
498	6012	HULA-A	1213	75	15
412	7225	HULA-A	1322	50	15
888	DOWN	HULA-A	1376	64	0
469	9445	HULA-B	15	30	15
430	7314	HULA-B	80	15	15
485	6012	HULA-B	111	130	15
130	9757	HULA-B	307	15	15
888	DOWN	HULA-B	320	43	0
57	2532	HULA-B	383	15	20
58	2532	HULA-B	482	15	20
121	7050	HULA-B	570	13	20
453	8896	HULA-B	601	40	15
107	6553	HULA-B	747	3	15
386	6374	HULA-B	766	15	15
888	DOWN	HULA-B	800	87	0
71	3187	HULA-B	907	13	20
76	3726	HULA-B	947	10	20
72	3187	HULA-B	1005	17	20
414	7225	HULA-B	1038	130	15
888	DOWN	HULA-B	1171	80	0
494	7304	HULA-B	1267	15	15
495	6142	HULA-B	1303	95	15
427	7310	HULA-B	1414	20	15
379	6071	COOK-A	20	10	15
463	9441	COOK-A	50	10	15
888	DOWN	COOK-A	71	118	0
129	9757	COOK-A	209	17	20
312	2272	COOK-A	242	15	15
42	1132	COOK-A	282	15	20
51	1748	COOK-A	334	17	20
448	8275	COOK-A	367	30	15
404	7225	COOK-A	413	10	15
314	2567	COOK-A	439	15	15
19	1056	COOK-A	509	14	20

397 6453	COOK-A	539	15	15
480 7225	COOK-A	580	15	15
369 5953	COOK-A	610	20	15
400 6542	COOK-A	651	10	20
416 7304	COOK-A	718	45	15
398 6453	COOK-A	780	10	15
405 7225	COOK-A	823	15	15
888 DOWN	COOK-A	858	68	0
142 9757	COOK-A	946	16	20
323 3726	COOK-A	1023	23	20
411 7225	COOK-A	1062	65	15
422 7304	COOK-A	1208	45	15
888 DOWN	COOK-A	1320	120	0
380 6071	COOK-B	15	30	15
381 6071	COOK-B	61	30	15
395 6451	COOK-B	107	10	15
50 1748	COOK-B	239	15	20
479 7225	COOK-B	270	100	15
407 7225	COOK-B	393	260	15
888 DOWN	COOK-B	726	75	0
421 7304	COOK-B	817	165	15
45 1132	COOK-B	1002	10	20
432 7314	COOK-B	1028	30	15
409 7225	COOK-B	1089	10	15
433 7314	COOK-B	1120	20	15
440 7506	COOK-B	1156	10	15
37 1056	COOK-B	1244	16	15
446 7837	COOK-B	1289	35	15
39 1056	COOK-B	1346	13	15
115 6553	COOK-B	1364	11	5
888 DOWN	COOK-B	1375	65	0
77 4774	INDI-A	27	14	15
464 9442	INDI-A	57	64	15
337 4524	INDI-A	137	30	15
344 4845	INDI-A	183	30	15
333 4035	INDI-A	223	30	5
888 DOWN	INDI-A	254	62	0
385 6280	INDI-A	360	30	15
97 5821	INDI-A	417	12	20
355 5329	INDI-A	445	20	15
362 5775	INDI-A	490	40	15
21 1056	INDI-A	550	16	20
478 7506	INDI-A	582	15	15
106 6553	INDI-A	687	13	20
302 470	INDI-A	716	45	15
108 6553	INDI-A	784	12	20
155 9845	INDI-A	825	16	20
376 6012	INDI-A	857	130	15
143 9757	INDI-A	1007	12	15

352	5329	INDI-A	1035	20	15
63	2532	INDI-A	1078	15	20
373	6012	INDI-A	1109	70	15
471	9446	INDI-A	1190	50	10
38	1056	INDI-A	1302	17	20
356	5329	INDI-A	1360	20	15
95	4774	INDI-A	1402	11	20
384	6280	INDI-A	1429	10	15
444	7641	BOSS-A	15	44	15
456	9363	BOSS-A	75	56	15
461	9434	BOSS-A	147	30	15
311	2124	BOSS-A	210	20	10
437	7506	BOSS-A	298	10	15
327	4035	BOSS-A	314	10	5
449	8275	BOSS-A	340	30	15
476	9794	BOSS-A	386	30	15
82	4774	BOSS-A	505	13	20
83	4774	BOSS-A	602	14	15
138	9757	BOSS-A	641	15	20
888	DOWN	BOSS-A	655	67	0
139	9757	BOSS-A	742	16	20
388	6391	BOSS-A	790	10	15
321	3310	BOSS-A	851	45	15
475	9783	BOSS-A	912	15	15
503	6738	BOSS-A	943	45	15
442	7506	BOSS-A	1005	200	15
888	DOWN	BOSS-A	1210	93	0
493	7304	BOSS-A	1319	20	15
428	7310	BOSS-A	1355	10	15
888	DOWN	BOSS-A	1367	73	0
126	9757	BOSS-B	10	16	15
96	5821	BOSS-B	87	13	20
49	1748	BOSS-B	141	17	20
329	4035	BOSS-B	164	10	5
56	2532	BOSS-B	189	10	14
371	6012	BOSS-B	215	71	15
15	1056	BOSS-B	310	14	20
888	DOWN	BOSS-B	324	66	0
16	1056	BOSS-B	410	16	20
101	6553	BOSS-B	452	14	20
68	3187	BOSS-B	507	12	15
477	9794	BOSS-B	535	15	15
415	7304	BOSS-B	566	15	15
70	3187	BOSS-B	604	17	15
507	7225	BOSS-B	637	260	15
486	9363	BOSS-B	913	10	15
30	1056	BOSS-B	940	11	15
439	7506	BOSS-B	1005	10	15
32	1056	BOSS-B	1040	16	20

888	DOWN	BOSS-B	1056	55	0
394	6451	BOSS-B	1127	15	15
113	6553	BOSS-B	1165	12	20
326	3726	BOSS-B	1198	165	20
304	712	BOSS-B	1379	15	15
153	9757	BOSS-B	1422	16	20

11	1056	LION-A	11	13	15
888	DOWN	LION-A	24	75	0
303	712	LION-A	115	30	15
336	4373	LION-A	205	45	15
472	9446	LION-A	261	35	10
387	6391	LION-A	330	30	15
307	1920	LION-A	376	15	15
350	5037	LION-A	400	15	6
354	5329	LION-A	431	20	15
345	4955	LION-A	500	10	15
408	7225	LION-A	526	10	15
418	7304	LION-A	594	70	15
305	712	LION-A	680	30	15
122	7050	LION-A	737	12	7
447	7837	LION-A	772	30	15
124	7050	LION-A	836	15	20
110	6553	LION-A	868	12	15
888	DOWN	LION-A	880	60	0
111	6553	LION-A	965	15	15
341	4845	LION-A	996	30	15
441	7506	LION-A	1042	120	15
339	4524	LION-A	1178	30	15
148	9757	LION-A	1223	16	13
888	DOWN	LION-A	1239	55	0
53	1748	LION-A	1314	16	20
47	1132	LION-A	1369	15	20
332	4035	LION-A	1390	10	5
40	1056	LION-A	1422	11	17

888	DOWN	LION-B	0	180	0
78	4774	LION-B	209	13	20
100	6553	LION-B	257	14	20
79	4774	LION-B	306	14	15
132	9757	LION-B	336	15	15
81	4774	LION-B	404	10	14
134	9757	LION-B	437	16	15
426	7310	LION-B	469	10	15
351	5329	LION-B	500	20	15
136	9757	LION-B	540	11	15
360	5775	LION-B	570	20	15
328	4035	LJON-B	596	10	5
23	1056	LION-B	635	11	15
98	5821	LION-B	693	13	20
25	1056	LION-B	735	16	15

353	5329	LION-B	767	50	15
27	1056	LION-B	837	15	15
89	4774	LION-B	875	13	15
888	DOWN	LION-B	888	63	0
91	4774	LION-B	971	14	18
331	4035	LION-B	991	10	5
450	8275	LION-B	1020	15	15
112	6553	LION-B	1063	12	20
146	9757	LION-B	1122	15	20
510	7225	LION-B	1153	85	15
460	9366	LION-B	1290	10	15
365	5775	LION-B	1320	40	15
383	6280	LION-B	1376	64	15
55	2532	GUAM-A	20	13	20
391	6394	GUAM-A	49	30	15
13	1056	GUAM-A	143	13	20
888	DOWN	GUAM-A	156	67	0
14	1056	GUAM-A	243	16	20
474	9521	GUAM-A	341	15	15
80	4774	GUAM-A	373	14	15
135	9757	GUAM-A	507	14	20
452	8701	GUAM-A	537	52	15
137	9757	GUAM-A	607	16	15
43	1132	GUAM-A	669	15	20
392	6394	GUAM-A	700	10	15
123	7050	GUAM-A	766	15	20
888	DOWN	GUAM-A	802	83	0
28	1056	GUAM-A	905	16	20
31	1056	GUAM-A	1007	14	15
488	3726	GUAM-A	1037	240	15
118	6790	GUAM-A	1300	13	20
319	3160	GUAM-A	1329	20	15
888	DOWN	GUAM-A	1372	68	0
318	3160	GUAM-B	15	20	15
363	5775	GUAM-B	51	20	15
502	6738	GUAM-B	118	115	15
888	DOWN	GUAM-B	316	91	0
370	5953	GUAM-B	423	20	15
317	3055	GUAM-B	489	45	15
436	7506	GUAM-B	550	250	15
429	7310	GUAM-B	827	20	15
490	3726	GUAM-B	960	40	15
888	DOWN	GUAM-B	1020	60	0
361	5775	GUAM-B	1110	20	15
458	9365	GUAM-B	1176	58	15
149	9757	GUAM-B	1254	16	20
888	DOWN	GUAM-B	1270	67	0
152	9757	GUAM-B	1357	13	20
378	6012	GUAM-B	1386	35	15

306	1864	PIKE-A	20	10	20
888	DOWN	PIKE-A	30	60	0
127	9757	PIKE-A	110	16	20
120	7050	PIKE-A	167	15	20
315	2941	PIKE-A	234	15	15
316	3028	PIKE-A	265	15	15
505	7225	PIKE-A	348	75	15
457	9364	PIKE-A	462	57	15
103	6553	PIKE-A	550	12	20
424	7310	PIKE-A	578	10	15
84	4774	PIKE-A	605	12	15
105	6553	PIKE-A	646	14	15
888	DOWN	PIKE-A	659	29	0
87	4774	PIKE-A	701	15	15
372	6012	PIKE-A	732	35	15
310	2124	PIKE-A	778	20	10
308	2124	PIKE-A	809	20	10
141	9757	PIKE-A	844	16	15
368	5953	PIKE-A	880	20	15
309	2124	PIKE-A	915	20	10
888	DOWN	PIKE-A	1000	111	0
348	5037	PIKE-A	1118	10	6
34	1056	PIKE-A	1142	15	12
459	9366	PIKE-A	1176	10	15
125	7050	PIKE-A	1225	15	20
93	4774	PIKE-A	1266	10	20
94	4774	PIKE-A	1360	15	15
497	1920	PIKE-A	1391	35	15
343	4845	REEF-A	15	10	15
473	9446	REEF-A	36	25	10
434	7506	REEF-A	86	34	15
128	9757	REEF-A	151	16	20
454	8896	REEF-A	202	40	15
322	3726	REEF-A	263	22	20
359	5775	REEF-A	310	20	15
18	1056	REEF-A	451	14	15
888	DOWN	REEF-A	481	76	0
104	6553	REEF-A	591	11	16
320	3160	REEF-A	618	20	15
85	4774	REEF-A	661	15	15
335	4373	REEF-A	731	15	15
451	8639	REEF-A	762	15	15
140	9757	REEF-A	803	15	20
44	1132	REEF-A	859	13	19
382	6142	REEF-A	888	82	15
888	DOWN	REEF-A	975	67	0
364	5775	REEF-A	1058	80	15
35	1056	REEF-A	1201	16	20
403	6738	REEF-A	1262	45	15

114 6553	REEF-A	1322	12	15
366 5775	REEF-A	1350	20	15
470 9445	REEF-A	1386	10	15
888 DOWN	REEF-A	1396	44	0

Unscheduled Supports

The following supports could not be scheduled.

Low Altitude Satellite Supports

5821(2)

3187

9757

6790

Medium/High Altitude Satellite Supports

7304

3055

3726

3726

6012

5037

5037

5037

7304

7304

7225

7225

7225

7304

6453

4524

9444

6012

6012

3055

4035

9434

4832

9444

4845

4035

4845

Bibliography

Air Force Satellite Control Facility. *AFSCF Training Manual.* SCF Orientation Course 203. Contract F04690-81-C-004 with Ford Aerospace & Communication Corporation. AFSCF, February 1983.

Arbabi, M. " Range Scheduling Automation", *IBM Technical Directions* , 10 (3): 57-62, (1984)

Arbabi , M, Garate, John, and Kocher, Donald. "Interactive Real Time Scheduling and Control", *Proceedings of the 1985 Summer Computer Simulation Conference*. 271-277. Chicago: Elsevier Science Publishers, 1985.

Arbabi, M . IBM Senior Research Analyst, IBM Federal Systems Division, Gathersburg, Md., Telephone Interview, 24 July 1992

Asker, James R. "Demand for Space Products, Services Grows at Healthy Rate". *Aviation Week & Space Technology*, 132 (22): 51 (May 28, 1991).

Baker,Bruce N. Introduction to Sequencing and Scheduling. New York: John Wiley & Sons, 1974.

Bodin, Lawrence, Golden, Bruce, Assad, Arjang, and Ball, Michael. "Routing and Scheduling of Vehicles and Crews - the State of the Art", *Computers and Operations Research*, 10 (2): 1-211 (1983)

Chambers, Ken. Technical Advisor, Range Scheduling Branch, 21 SOPS, Onizuka AFB, Personal Interview, 17 June 1992.

Chambers, Ken. Technical Advisor, Range Scheduling Branch, 21 SOPS, Onizuka AFB, Telephone Interview, 24 October 1992.

Chambers, Ken. Technical Advisor, Range Scheduling Branch, 21 SOPS, Onizuka AFB, Telephone Interview, 15 January, 1993.

Foster, Capt Gary. " Automating the Weekly Flight Scheduling Process at the USAF Test Pilot School", Ms Thesis, AFIT/ENS/92M-110. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, March 1992

French, Simon. Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop. London: Ellis Harwood, Ltd, 1982.

Nemhauser, George and Wolsey, Laurence. Integer and Combinatorial Optimization. New York: John Wiley & Sons, 1988.

Parker, Gary R, and Rardin, Ronald L.. "An Overview of Complexity theory in Discrete Optimizations: Part I. Concepts" *IIE Transactions*, 14 (1): 3-10 (March 1982a).

Parker, Gary R Gary and Rardin, Ronald L.. "An Overview of Complexity theory in Discrete Optimizations: Part II. Results and Implications" *IIE Transactions*, 14 (2): 83-89 (June 1982b).

Schultz, 1Lt Greg. SRS Project Officer, HQ SMC/CWD, Telephone Interview, 11 February 1993.

Solomon, Marius. "Algorithms for the Vehicle-Routing and Scheduling Problems with Time Window Constraints", *Operations Research*, 35 (2): 254-265 (1987)

Syslo, Deo. Discrete Optimization Algorithms. New York: Prentice Hall Inc., 1983

Wong, Francis. Supervisor of Planning, Space Operations Flight, 21 SOPS, Onizuka AFB, Personal Interview, 18 June 1992.

Wong, Francis. Supervisor of Planning, Space Operations Flight, 21 SOPS, Onizuka AFB, Telephone Interview, 29 January 1993.

Vita

Captain Timothy D. Gooley was born October 19, 1962 in Glens Falls, New York. He graduated from Greenwood Central School in Greenwood, New York in 1981, and attended the U.S. Air Force Academy, graduating with a Bachelor of Science in Operations Research in 1985. Upon graduation, he was assigned to Headquarters Air Training Command at Randolph AFB. While at HQ ATC, he served as a maintenance analyst in the Maintenance Analysis Division and a training effectiveness analyst in the Command Analysis Division. He received a Masters in Business Administration from St. Mary's University, San Antonio, Texas in 1991. He entered the School of Engineering, Air Force Institute of Technology, in August 1991.

Permanent address: 4441 Route 224
Montour Falls, NY 14865
c/o Donald Gooley